Visual Symphony with Jitter: Cymatic Waters

Self-directed project

Acacia Williams

CART 346: Digital Sound I: Theory And Practice of Real-time Audio

Department of Computation Art, Concordia University

Dr. Gabriel Vigliensoni

December 4, 2024

Abstract

Visual Symphony with Jitter: Cymatic Waters is an immersive audio-visual project that reimagines sound as an interactive and explorative medium. The patch, created in Max MSP and Jitter, combines six water-themed audio sources with dynamic visual representations inspired by cymatics, the study of how sound vibrations produce patterns in physical matter. At the core of the system lies a nodes-based interface, where each node represents a sound, enabling users to seamlessly blend, isolate, or overlap sounds by manipulating a central control point. This intuitive spatial interface offers unique opportunities for real-time sound exploration and customization.

The audio component supports drag-and-drop functionality for loading sounds and incorporates features such as volume, pitch, panning, frequency modulation, and phasing delay. Randomization and reset buttons further enhance the exploratory nature of the patch. The visual component leverages Jitter to simulate cymatic patterns in a jit.world environment, dynamically linking audio data to visual parameters such as vibration intensity and pattern complexity. This integration is powered by snapshot~ objects capturing real-time amplitude data.

The result is an innovative tool that bridges auditory and visual creativity, offering structured control alongside playful experimentation. Suitable for performers, sound designers, and meditative users, the patch invites participants to navigate soundscapes as tangible landscapes, fostering deeper connections between what they hear and see. By blending artistic expression with intuitive interactivity, Cymatic Waters achieves a unique synthesis of technology, creativity, and sensory engagement.

Introduction

Background:

In the field of digital sound production, the convergence of auditory and visual elements has increasingly become a powerful tool for creative expression. One fascinating area of exploration lies in cymatics, the study of how sound vibrations influence physical matter like sand or water, to produce visible patterns. These natural phenomena, observed in mediums such as water or sand, reveal the profound relationship between sound and its physical effects. Translating this concept into a digital medium opens up new possibilities for interaction and creativity, allowing users to explore sound not only as an auditory experience but also as a tangible, visual phenomenon. Cymatics reveals how sound frequencies, such as six water sounds, can influence and create patterns in matter, illustrating the profound connection between auditory and visual phenomena.

Influences:

The project was inspired by a desire to create an immersive audio-visual experience that goes beyond traditional listening. Initial explorations into tranquil sounds led to research on Jitter's capabilities. Discovering a cymatics simulation video on YouTube showcased the mesmerizing patterns formed by sound vibrations, providing the foundation for the project. Additional inspiration came from Ableton's Learn Synths platform, highlighting the importance of visual feedback in sound manipulation.

Objectives:

The primary objective was to create an Interactive Sound Mixer with Modulation Features, combining intuitive sound blending, advanced audio processing, and cymatic-inspired visuals. The system uses a node-based interface allowing users to navigate and manipulate six water-themed audio sources. Each node represents a distinct sound, and users can blend, isolate, or overlap sounds by moving a control point.

Modulation features like pitch control, panning, frequency modulation, and phasing delay provide creative flexibility. Randomization and reset buttons enhance exploratory potential.

Complementing the mixer is a cymatic-inspired visual component rendered in real time using Jitter. Visuals respond to audio data, translating sound characteristics into evolving patterns. This interplay of motion, sound, and visuals reinforces the connection between what users hear and see, enriching the experience. Together, the Interactive Sound Mixer and the cymatics simulation form a cohesive system where auditory and visual elements complement and amplify one another, fostering deeper sensory engagement and creative expression.

Project Description

Concept Overview

The project, Visual Symphony with Jitter: Cymatic Waters, is an exploration of sound as both an auditory and visual medium, creating an immersive and interactive audio-visual environment. At its core, the project aims to bridge the natural phenomenon of cymatics—patterns formed by sound vibrations in matter—with Max and Jitter to make sound tangible and navigable. With an interactive node-based interface, users can seamlessly interact with six water-themed audio sources and manipulate the nodes to craft personalized soundscapes, exploring the blending and intensity of each sound tied to each node. This patch is more than just a mixer; it's an Interactive Sound Mixer with Modulation Features designed to give users control over multiple parameters like pitch, panning, frequency modulation and phasing delay. The phaser and frequency modulation were added as nodes to explore these sound modulation parameters, relative to the other sounds. The visuals, inspired by real-world cymatics, enhance the auditory experience, providing immediate feedback through mesmerizing, evolving patterns that directly respond to the sound. This combination invites users to not only hear sound but also feel and see it, creating a space for meditative exploration, creative experimentation, and artistic expression.

Sonorities Explored

The patch employs six water-themed audio sources, including sounds like paddle strokes, fountains, running faucets, water drops, rain, and splashes. Sounds are loaded through a drag-and-drop interface (to encourage sound exploration) and visualized via waveform displays. Users blend sounds by moving the control point within the node system, with proximity to a node determining the intensity of its associated sound. The result is a dynamic sound environment where users can explore water-inspired textures, from calm ripples to layered compositions. Additional sound processing features include:

- Pitch Modulation: pitch shifts using sig~ objects connected to groove~.
- Panning: Managed using pan2 objects, and number object giving spatial depth to the sounds.
- Frequency Modulation and Phasing Delay: Introduced through tapin~, tapout~, and cycle~ objects, adding temporal and tonal variety. Implemented as a node, via unpack.
- Randomization: Buttons to randomize or reset pan and volume settings.

Algorithmic and Logical Structures

Nodes Object Implementation

The node interface serves as the central control mechanism, mapping each sound to a node through unpack and live.gain~. This weighting system (nodes object outputs values ranging from 0 to 1) allows for smooth crossfading between sounds and effects. When centered on a node, only that sound is heard. Moving between nodes blends the sounds smoothly, while overlapping nodes create complex, multi-layered outputs. The proximity of the control point within the node space determines the intensity of each sound, enabling smooth blending and transitions. Additional parameters like phasing and frequency modulation are also linked to nodes, enhancing the interactive nature of the patch.

Audio Processing

- **Sampling:** Audio samples are loaded into buffer~ objects via drag-and-drop.
- **Playback**: groove~ objects read from buffers, controlling playback speed.

- **Sound manipulation:**
    - **Pitch**: Controlled through sig~, connected to each groove~
    - **Panning**: Managed through pan2 objects for stereo spatialization.
    - **Phasing Delay (node)**: Implemented using tapin~ and tapout~ objects.
    - **Frequency Modulation (node):** Applied through cycle~ objects.
    - **Randomization and Reset:** send and receive objects introduce randomness with reset options.

The visual component simulates cymatic patterns using Jitter, translating audio data into evolving animations. For the particle system, a large number of particles (100,000) are created using jit.noise, simulating sand particles on a vibrating plate. They are visualized as points through jit.gl.mesh. The z-axis is flattened to simulate a two-dimensional plane (particles are placed onto 2D plane to simulate the flat surface of water). Particles are mapped within a coordinate system ranging from -1 to 1, centered using jit.map.

The cymatics algorithm, uses a simplified Chladni pattern formula in jit.gen:

force=(sin(πmx)sin(πny)+sin(πnx)sin(πmy))×v .

Parameters: **m** and **n** (frequencies), **v** (vibration intensity influenced by audio input). Particle positions are stored in a jit.matrix object and updated continuously using jit.gen, where the algorithm calculates the force acting on each particle.

Audio-Visual Synchronization:

Output from live.gain~ objects is captured via snapshot~, scaled, and controls m, n, and v in real time.

Technologies and Tools

Software:

- **Max MSP**: Foundation for building the patch.
- **Jitter:** Used for graphics and video processing.

Hardware:

- **Laptop:** Development and testing.

External Resources:

While the project was built from the ground up, initial concepts and techniques were learned from tutorial videos on YouTube. These resources provided foundational knowledge on using the nodes object and simulating cymatics in Max MSP and Jitter. The audio patch underwent significant transformation as I implemented a drag and drop, buffer and groove playback system, and integrated the best sound modulation elements of my prior assignments and expanded upon them to create my personalized unique and interactive sound manipulation system. Initially, my assignments introduced me to foundational techniques such as basic audio playback, modulation, and spatial effects. Inspired by these, I reimagined and restructured the patch into a more advanced and dynamic Interactive Sound Mixer with Modulation Features. The Jitter component of the project was equally transformed through the integration of real-time audio data into the cymatics simulation. Initially, the Jitter patch was designed to simulate cymatic patterns based on static or manually controlled parameters. I expanded this concept by dynamically linking the audio output to the visual representation, creating a cohesive and immersive audio-visual experience, powered with my node based sound mixer.

- Youtube. 2022. "Nodes - Experimental Audio Crossfading Patch, Max MSP Tutorial." Accessed December 4, 2024. Nodes - Experimental Audio Crossfading Patch - Helpful Objects to Know - …

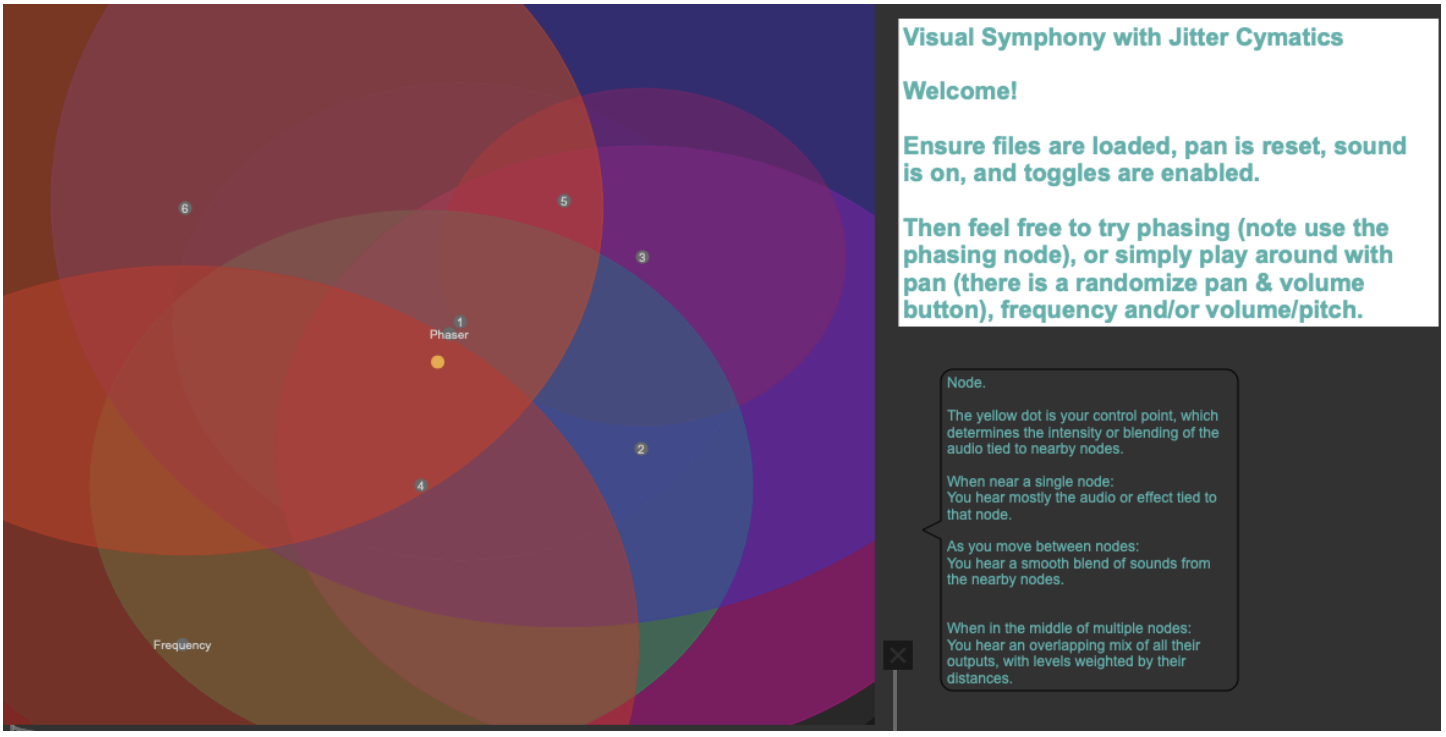
- Youtube. 2022. "Cymatics - Max/MSP Tutorial."Accessed December 4, 2024.

  Cymatics - Max/MSP Tutorial

Implementation Details

Patch Structure

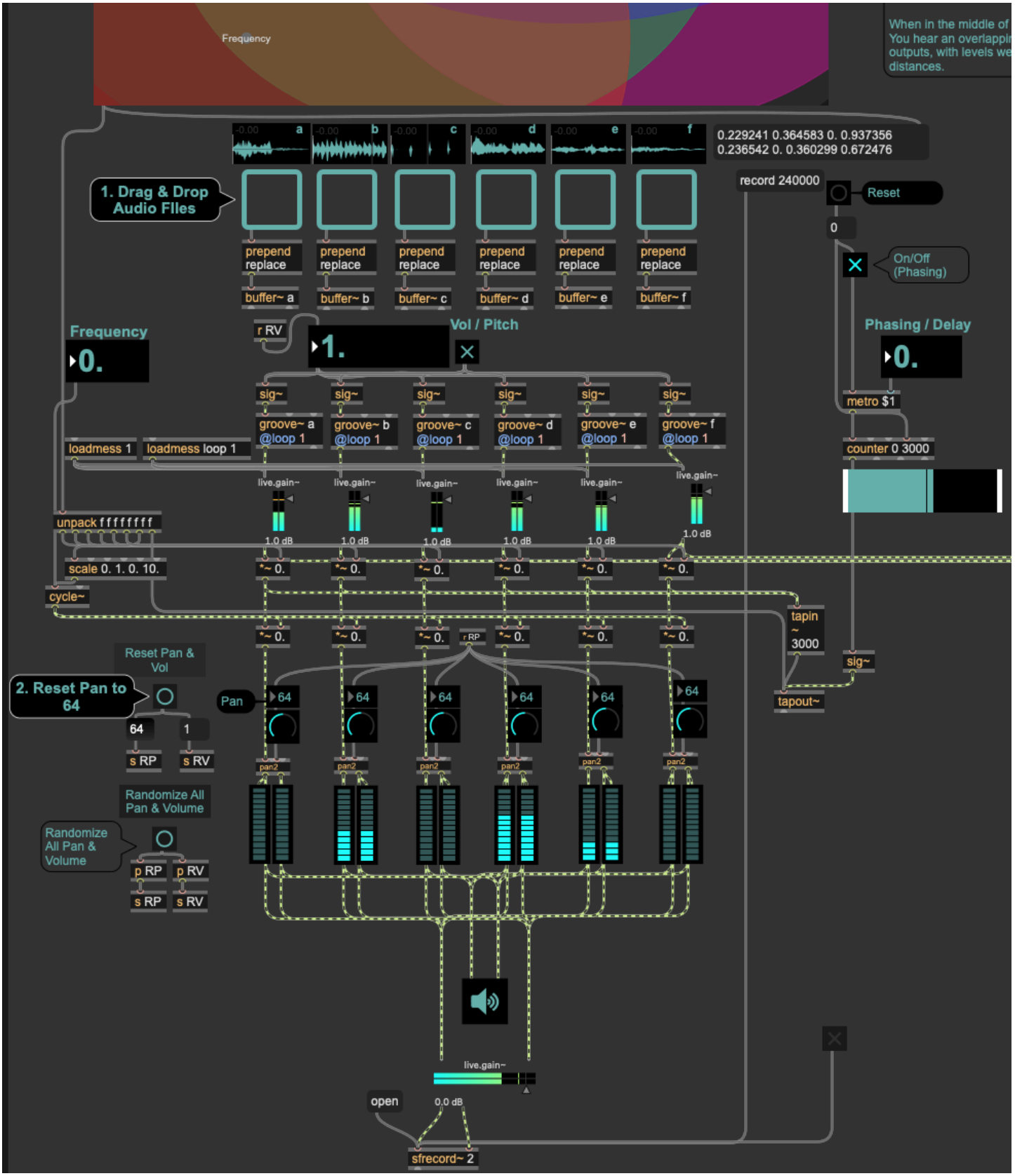
The patch was structured to seamlessly integrate both audio and visual components, creating an immersive experience. For the nodes initialization and audio mapping, the nodes object was configured with six nodes representing water sound samples, alongside additional nodes for controlling sound effects such as phasing delay and frequency modulation. The 'circle' display knob was enabled for visual interaction, allowing users to intuitively navigate and blend sounds. The nodes object outputs weighted values based on the control point's proximity to each node, and these values were unpacked using the unpack object for individual processing and routing to the audio and effects systems.



In the audio processing component, sampling was handled by loading the six water-themed audio samples into buffer~ objects, utilizing drag-and-drop functionality to encourage sound experimentation and user convenience.

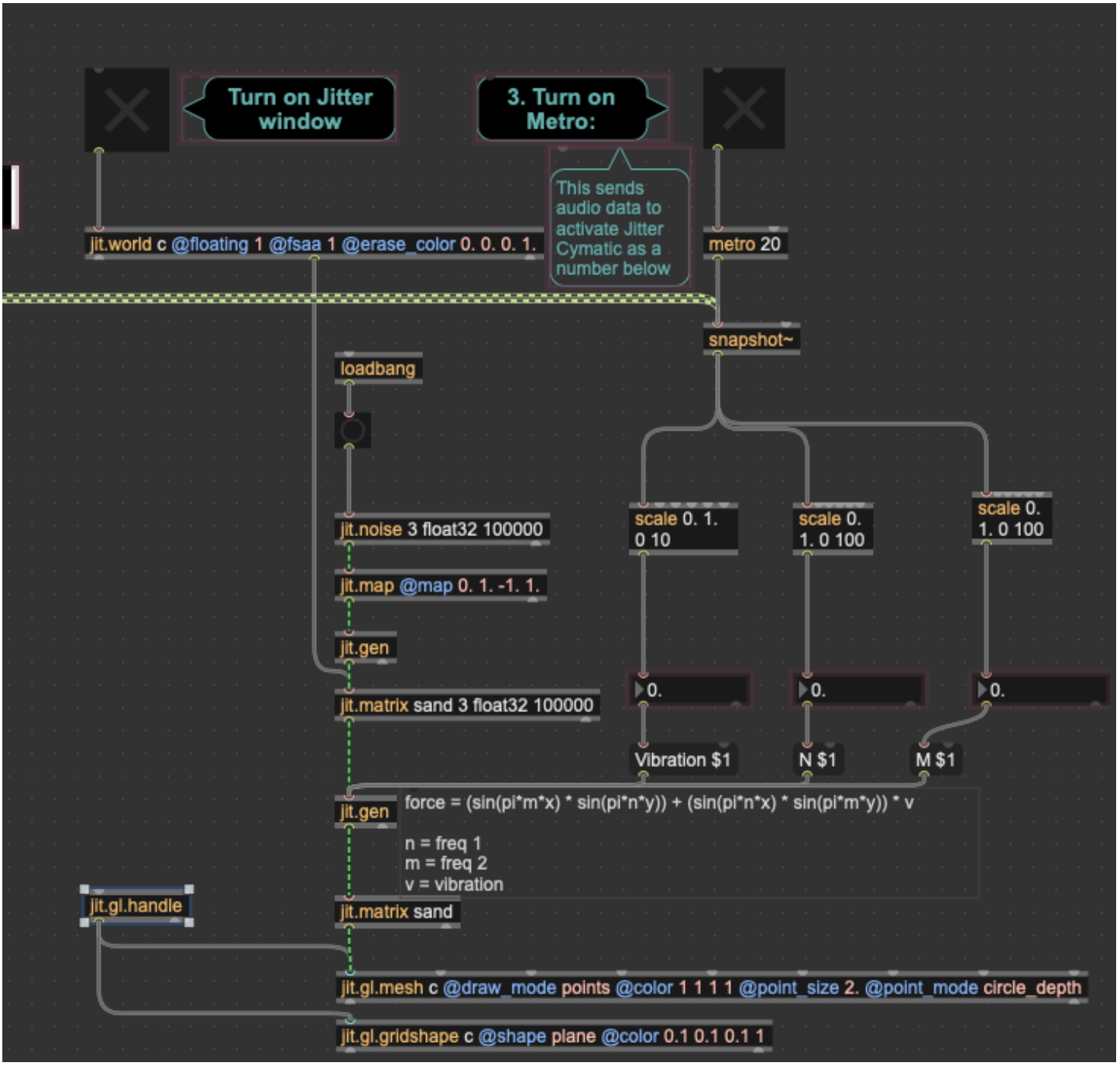
For playback, groove~ objects were employed to read from the buffers and control playback speed. Volume control was achieved by connecting live.gain~ objects to the outputs of groove~, allowing for individual volume adjustments.

The patch included several sound manipulation features. Phasing delay was implemented using tapin~ and tapout~, allowing for adjustable delay times controlled by node proximity. Frequency and pitch were adjusted using sig~ and line~ objects to manipulate playback speed and pitch based on node weights. Pan control, providing spatial audio effects, was managed with pan2 objects; using *~ and +~ operations, the audio signals were distributed across stereo channels, with dials and visual meters offering precise control. Randomization was introduced through send and receive objects, which added randomness to pan and volume settings, with controls to randomize or reset these parameters.



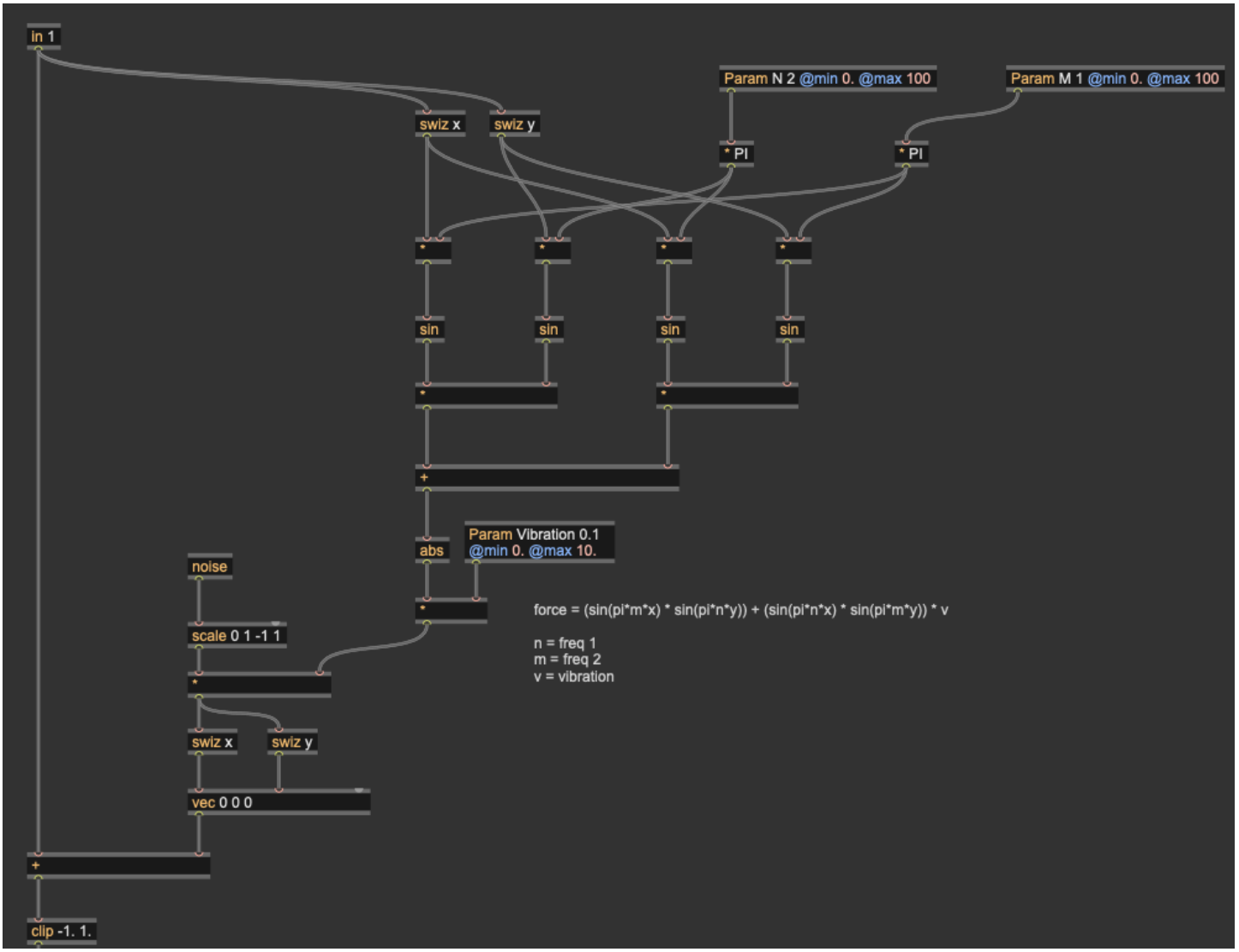
For the cymatics component in Jitter, a jit.world object was created to render the visual output in real time, featuring anti-aliasing and customizable background colors for enhanced visual quality. The particle system was established by generating a matrix of particles with jit.noise, simulating sand particles on a vibrating plate. Position mapping was handled by jit.map, which adjusted particle positions to center them

within the viewing window. The z-axis was flattened using jit.gen by setting all z-values to zero, effectively simulating a two-dimensional surface.



The cymatics algorithm implementation involved several key steps. In jit.gen, parameters for m, n, and v were included and dynamically controlled by audio input. The force acting on each particle was calculated using the formula: force = (sin(πmx)sin(πny) + sin(πnx)sin(πmy)) × v, where m and n represent frequencies controlling the patterns, and v is the vibration intensity influenced by the audio input. Random movement was introduced using noise() functions to add natural variability to particle movements. A feedback loop was

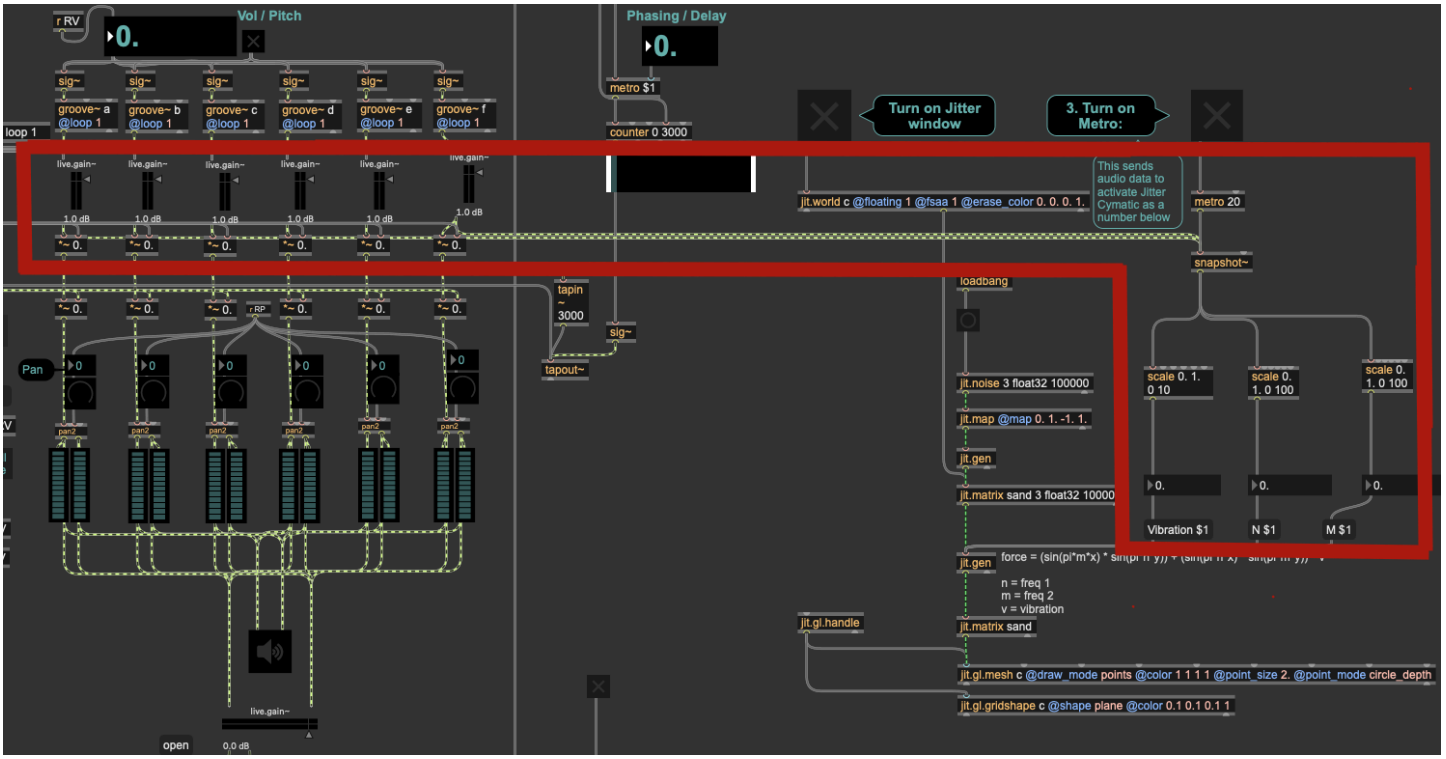
established by continuously updating particle positions, simulating the ongoing vibration of the plate.



Visualization was achieved by rendering the particles using jit.gl.mesh in point draw mode. A plane representing the vibrating surface was created with jit.gl.gridshape, and jit.gl.handle allowed for adjustments to the viewing angle, giving users the ability to explore the visuals from different perspectives.

For audio-visual integration, the connection between audio and visuals was implemented after trial and error. The output from live.gain~ objects was sent to snapshot~, then scaled appropriately to match the expected input ranges of the visual parameters. These scaled values controlled the m, n, and v parameters in the cymatics algorithm within jit.gen, ensuring that changes in audio directly influenced the visuals. This tight coupling

between sound and visuals created a cohesive and responsive audio-visual experience.



Technical Challenges

While working and testing with Jitter attributes, I experienced Max crashing (this was a first). I tested jitter in multiple patches and ran into issues when a jitter object had the same name in two patches, which required changing the name on one, saving and resetting both patches. Setting the attributes to arguments when resetting the patch was a key skill I learned with Jitter (to ensure the path ran smoothly upon opening it). Synchronization between audio and visuals posed another challenge, addressed by smoothing and scaling amplitude data captured via snapshot~, after trial and error troubleshooting. Integrating advanced audio modulation features like phasing delay and frequency modulation without introducing instability required isolating effects processing and prioritizing system stability, ensuring both creative depth and performance.

Results and Discussion

Project Outcomes:

I think the project successfully achieved its objectives of creating an immersive, interactive audio-visual experience that digitally simulates the phenomenon of water cymatics. By combining sound manipulation with real-time visual feedback, users were provided with a dynamic environment for creative exploration. The nodes interface enabled intuitive sound blending, while the visual cymatic patterns enhanced the sensory connection between sound and visuals. The incorporation of advanced sound manipulation features, such as phasing delay, frequency modulation, pitch adjustments, and panning, added layers of complexity to the user experience. These features allowed users to personalize their soundscapes while maintaining the meditative and experimental qualities of the patch. Additionally, the randomization functions encouraged playful discovery, meeting the goal of fostering both structured creativity and spontaneous exploration. Because this was a hands-on immersive project, I wish others were able to experiment with it, to appreciate the full effect.

Sound Analysis:

The blending of six water-themed sounds through the nodes interface created a serene and immersive auditory experience, often described as both relaxing and evocative. Each sound maintained its natural, high-quality characteristics, even when subjected to extensive modulation. The node-based blending ensured smooth transitions and overlapping textures, allowing users to seamlessly navigate between isolated sounds and intricate mixes. Manipulation tools added depth and dimension, enabling users to shape the soundscape with precision and creativity. The randomization features injected an element of surprise, often producing unexpected but cohesive results that enhanced the exploratory nature of the patch. Mapping of sound parameters to the cymatic simulation, elevated the overall experience, making the project not only an overall technical success but also an engaging sensory exploration. Although the cymatics-inspired visuals effectively complemented the audio, occasional performance limitations, particularly with rendering large particle systems

could enhance the experience.

Conclusion

Summary

This project highlights the transformative power of sound as a medium for immersive interaction and creative expression. By integrating spatial sound manipulation with visually engaging cymatics simulations, the project invites users to explore and shape their sensory environment actively. The core sound component—designed as an Interactive Sound Mixer with Modulation Features—served as the foundation, offering users an array of tools to blend, modulate, and personalize six water-themed audio sources. Features such as pitch control, panning, phasing delay, and frequency modulation empowered users to craft intricate soundscapes while maintaining a sense of playfulness through randomization and reset options. The seamless integration of these audio manipulations with Jitter's dynamic cymatics visuals reinforced the connection between what is heard and what is seen, elevating the experience to one that is meditative and emotionally resonant.

Future Directions

Building on the success of this project, future iterations could further emphasize and enhance the sound component. For instance, adding live audio input would enable users to bring their own sounds into the mix, driving both auditory and visual elements in real time. Expanding the modulation features with more advanced effects, such as granular synthesis or spectral filtering, could provide users with even greater creative control. Gesture controls or MIDI inputs could be integrated to offer diverse methods for interacting with the soundscape, making the system more intuitive and performance-ready. On the visual side, incorporating more intricate cymatics patterns or exploring 3D representations of sound waves could deepen the sensory engagement. During the demonstration, I felt like rendering was an issue due to my laptop memory or GPU.

Handling rapid changes in node positions while managing high particle counts required leveraging Max MSP and Jitter's efficient signal processing and GPU acceleration. Performance optimization would ensure the patch runs smoothly across a wider range of hardware setups, making it more accessible.

References

Patch Audio Files:

- * water fountain by amlw097 – https://freesound.org/s/754944/ – License: Attribution NonCommercial 4.0
- * Running faucet by amlw097 – https://freesound.org/s/754942/ – License: Attribution NonCommercial 4.0
- * Water_Paddle_med_001.wav by EpicWizard – https://freesound.org/s/316576/ – License: Creative Commons 0
- * Water 16.aif by carlmartin – https://freesound.org/s/237260/ – License: Creative Commons 0
- * Soothing Waterdrop Click.wav by MATRIXXX_ – https://freesound.org/s/702806/ – License: Creative Commons 0
- * RAIN STICK A INCIDENTAL RATTLE 02.wav by sandyrb – https://freesound.org/s/86356/ – License: Attribution 4.0

Similar projects:

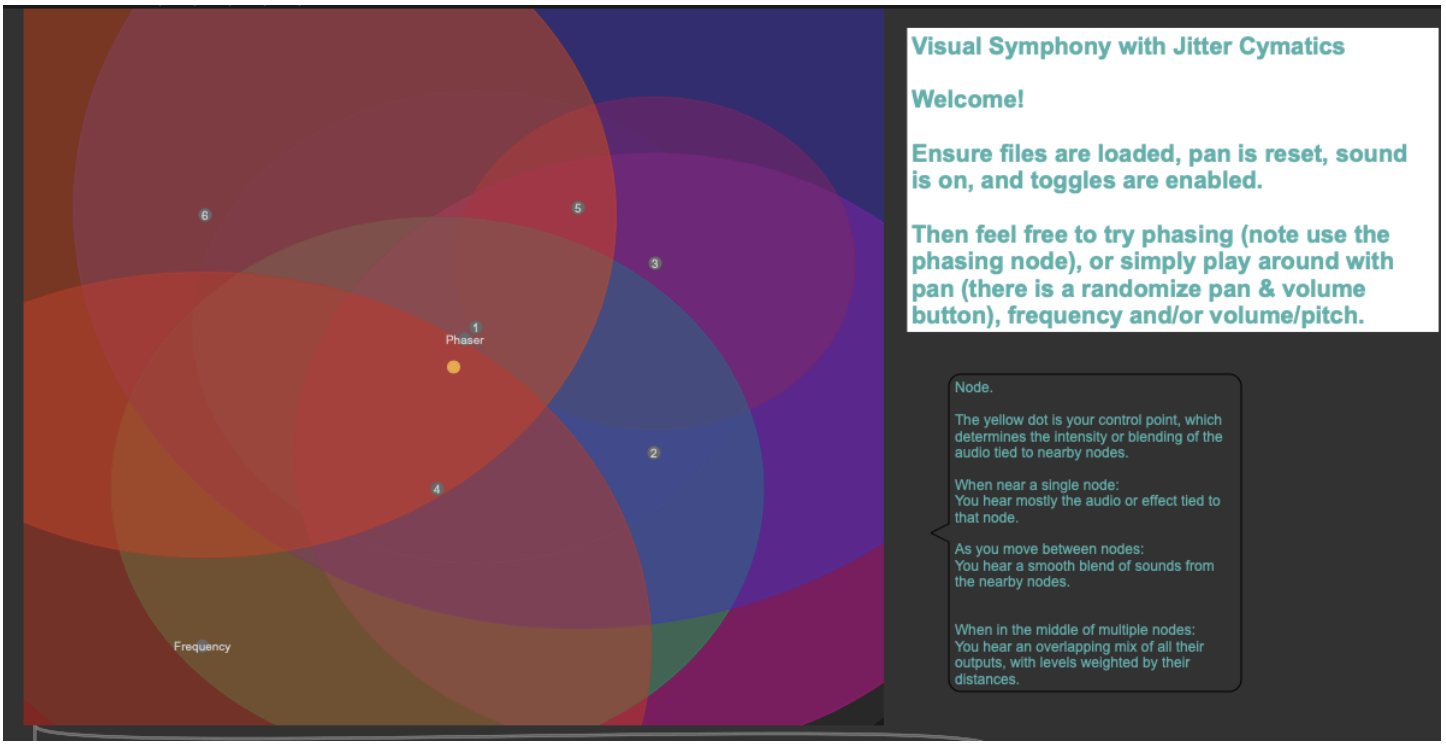
- Youtube. 2022. "Nodes - Experimental Audio Crossfading Patch, Max MSP Tutorial." Accessed December 4, 2024. Nodes - Experimental Audio Crossfading Patch - Helpful Objects to Know - …
- Youtube. 2022. "Cymatics - Max/MSP Tutorial."Accessed December 4, 2024. Cymatics - Max/MSP Tutorial

Appendix

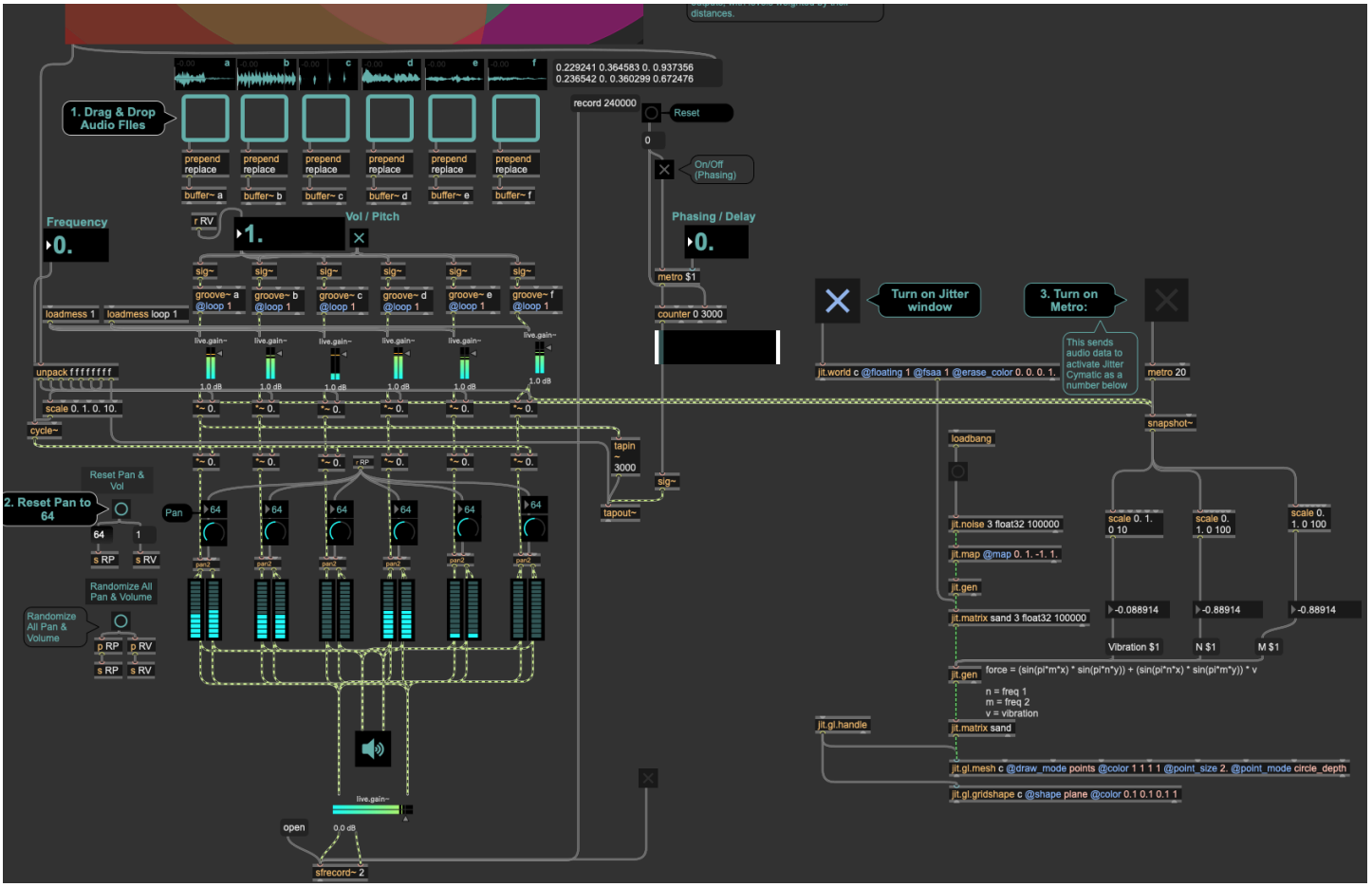
Slides

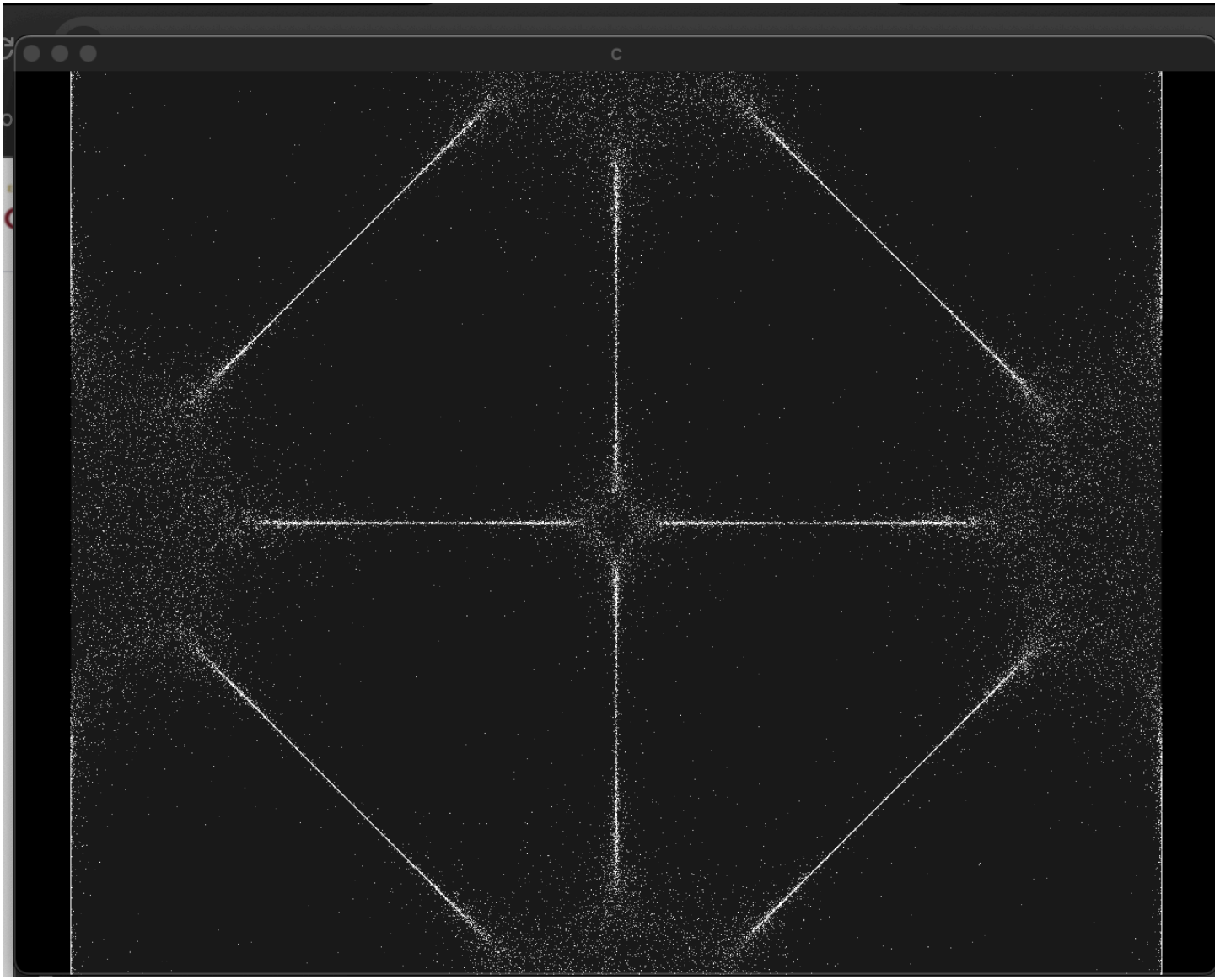
- https://www.canva.com/design/DAGXsKBDQSI/C61fCPPrHLgSw6n9reFNcg/view?utm_content=DAGXsKBDQSI&utm_campaign=designshare&utm_medium=link&utm_source=editor

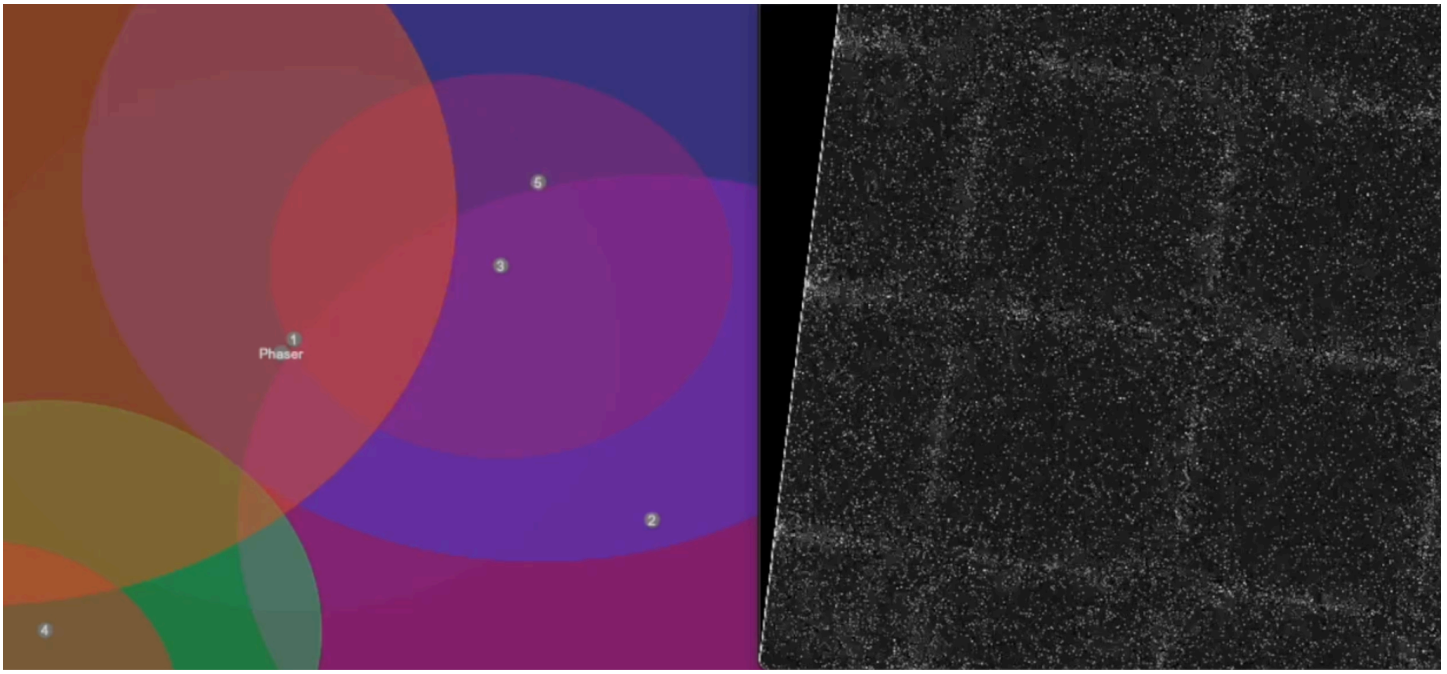
Additional Diagrams



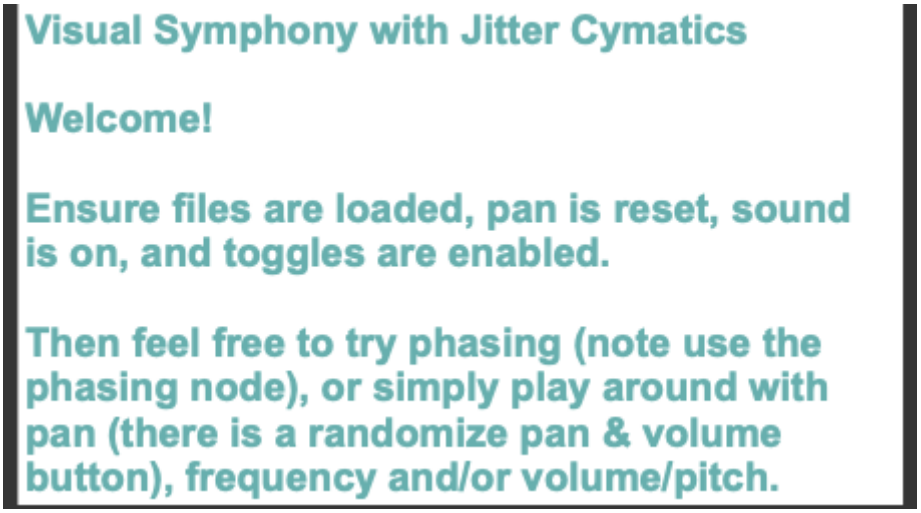
node



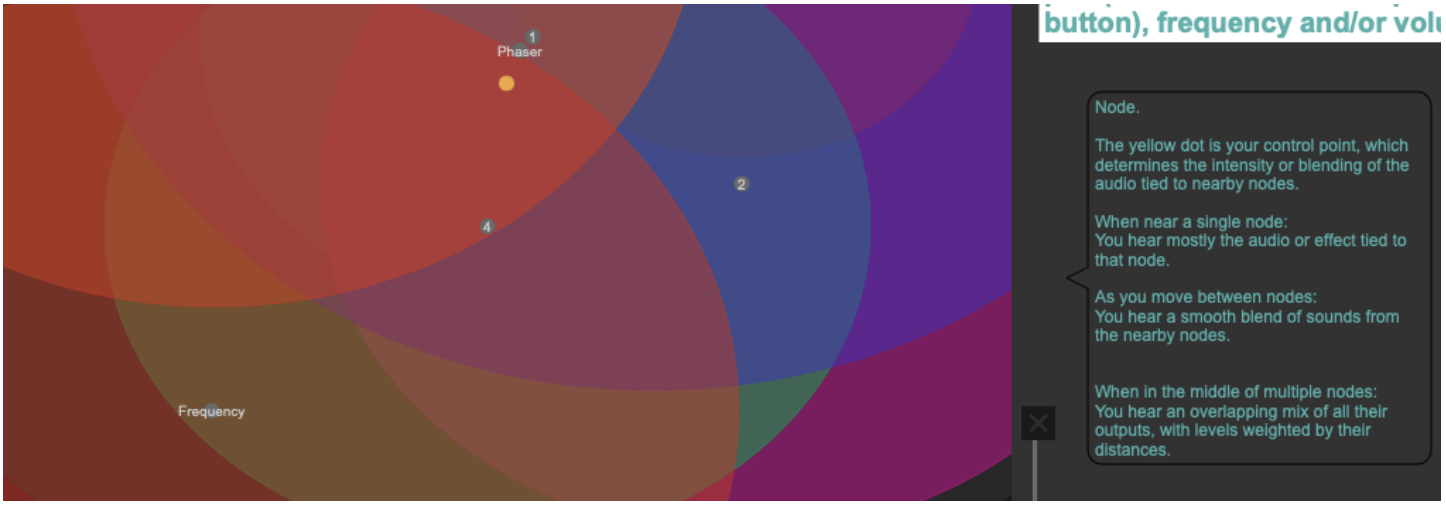




Instructions to Run Patch:

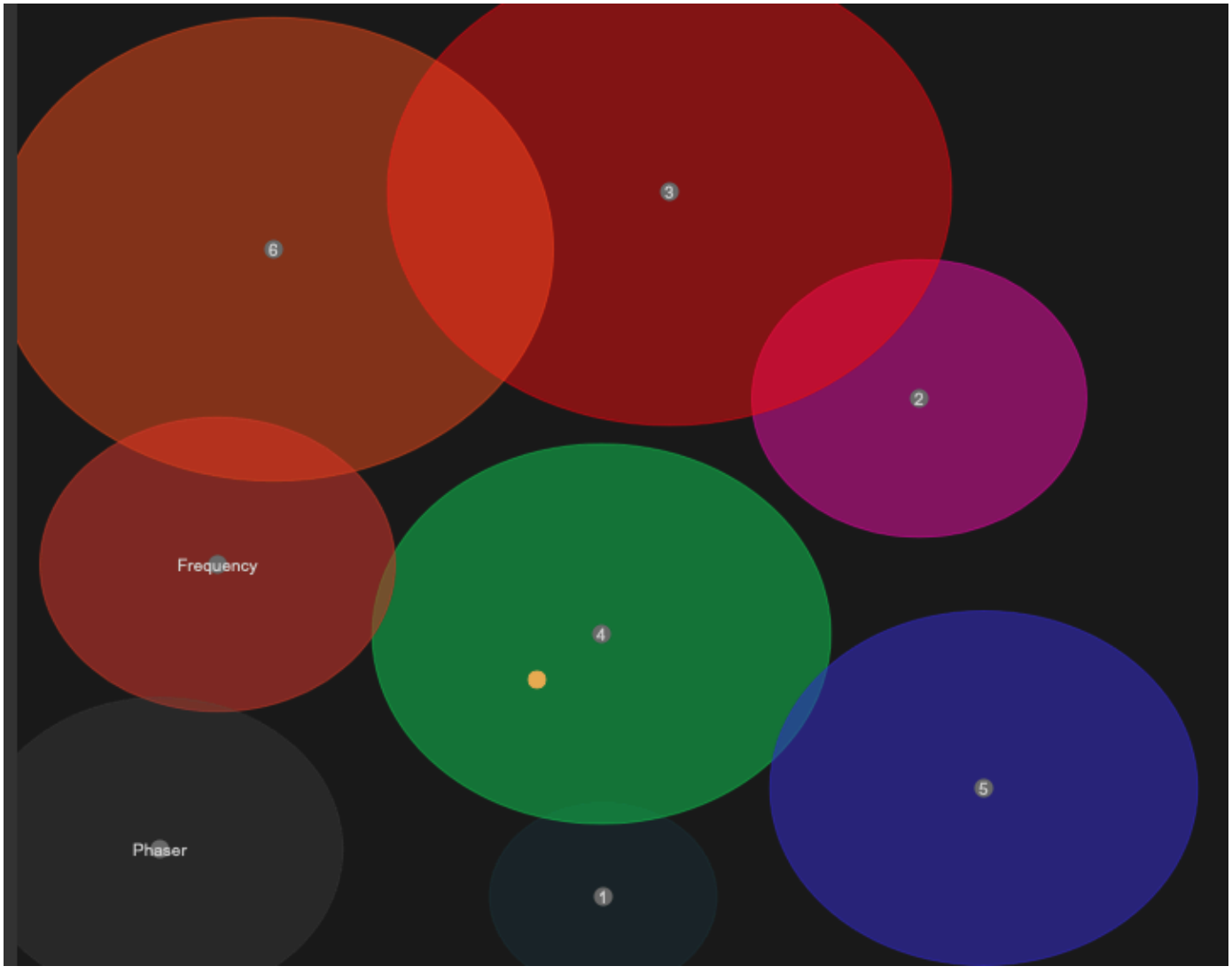


Now you can experiment with the sounds.



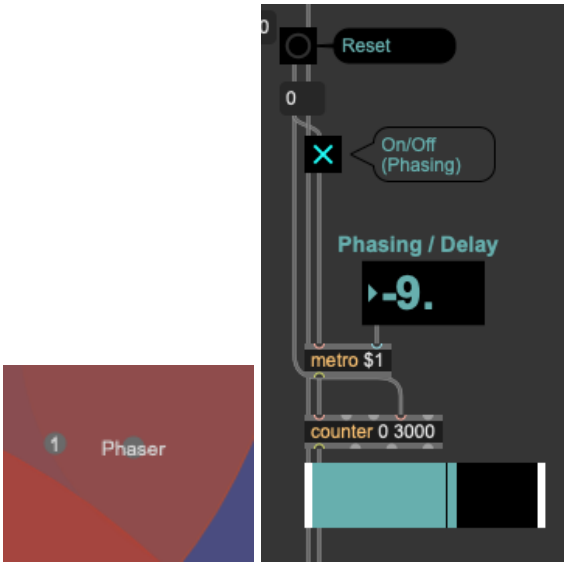
Adjust the nodes:

- The yellow dot is your control point, which determines the intensity or blending of the audio tied to nearby nodes.
- When near a single node: You hear mostly the audio or effect tied to that node.
- As you move between nodes: You hear a smooth blend of sounds from the nearby nodes.
- When in the middle of multiple nodes: You hear an overlapping mix of all their outputs, with levels weighted by their distances.

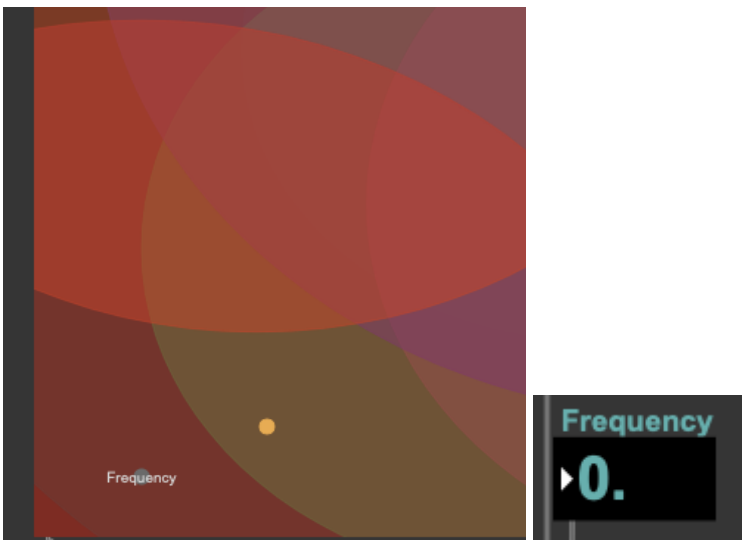


Randomize All Pan & Volume: Just click the button

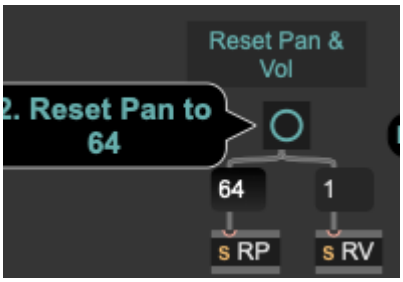
Try the Phaser Node: See how it affects the other sound nodes. Adjust it. Visualize it in the slider (at different speeds).



Try the Frequency Node: See how it affects the other sound nodes.



Or Reset! Just click the button to go back to the basics.



→