Creative Coding for Sound and Performance (Max)

Instructor: John MacCallum, PhD

Course Description

This course offers a hands-on introduction to interactive and generative audio in Max—a visual programming language for interactive sound and multimedia. We will examine how Max handles data and timing, and how to design patches for real-time sound synthesis, transformation, and performance.

Over the semester, students will work through focused studies leading to more advanced systems: phasor-driven timing, step sequencers, granular and FM-based processes, patches shaped by spectral information, and introductory multichannel techniques. Throughout, emphasis is placed on building patches that are clear, debuggable, and reliable in performance contexts.

We will work step by step, aiming for a practical understanding of how these systems behave and how to shape them into tools for artistic use.

Approach

The course emphasizes hands-on making and close examination of how patches behave in real time. Instruction combines demonstration, guided building, and individual experimentation. Students are encouraged to pay attention to the details of how data moves through a patch, to build systems incrementally, and to return to earlier work with increasing clarity. The aim is for each student to leave with a functional and well-understood system that can serve their ongoing practice.

Learning Goals

Students completing this course will be able to:

- Build modular Max patches that clearly separate control, timing, and audio processes.
- Use Max's message and signal systems to route, transform, and organize data.
- Employ phasors and continuous control signals to drive timing, tempo curves, and simple rhythmic structures.
- Develop basic synthesis processes, including envelope-based, granular, and simple FM approaches.
- Use spectral peak information from analysis objects to shape timbre or guide generative processes.
- Set up foundational multichannel and spatial routing using available Max objects.
- Use OSC or other communication protocols to connect Max with external devices or software.
- Prepare patches for performance by testing, documenting, and considering reliability and failure modes.
- Present and discuss technical and artistic decisions with clarity and specificity.

Course Structure — Thematic Units

1. Orientation: Messages, Signals, and Debugging

- Navigation in Max: objects, inlets/outlets, patching, and editing.
- What flows across a patch cord and how to inspect it.
- Ordering and control flow; Max scheduler vs. audio scheduler.
- Messages versus signals and when each is appropriate.
- Simple, single-function patches as building blocks.

2. Time and Timing: Phasors and Sequencing

- Phasor~ as a continuous representation of time.
- Relating phasors to tempo and beat-based thinking.
- Building a basic step sequencer.
- Converting between pulses and ramps; working with `line~` and `metro`.
- Creating tempo curves and simple accelerando/ritardando shapes.

3. Synthesis, Envelopes, and Polyphony

- Designing envelopes using objects such as `function`.
- Constructing a simple voice inside `poly~` (oscillator + envelope).
- Extending this voice into a basic granular-style process.
- Managing pitch, duration, and envelope data using `coll` or `dict`.
- Structuring patches so individual components can evolve over time.

4. Spectral Information and Timbre

- Overview of spectral analysis objects for obtaining peak information.
- Using spectral features to guide pitch, filtering, or textural change.
- Approaches to timbral "migration" based on weighted selection from spectra.
- Integrating these ideas with earlier granular and envelope work.

5. Routing, Space, and Interaction

- Fundamentals of audio routing (`send~/receive~`, `matrix~`).
- Introduction to multichannel techniques with `mc.*` objects.
- Recording and playback strategies using `buffer~`.
- Basic use of OSC for communicating with external tools or devices.
- Situating these elements in performance or installation contexts.

6. Performance Systems and Reliability

- Evaluating a patch for performance readiness.
- Strategies for testing and debugging real-time systems.
- Naming, commenting, and organizing patches for long-term use.
- Considering edge cases and planning for failure modes.
- Preparing and presenting a final project publicly.

Assignments & Evaluation

Weekly Patch Labs — 30%
Short exercises focusing on specific ideas introduced in class. Each lab is designed to reinforce a technical concept or method.

Prototype System (Mid-Semester) — 20%
An exploratory patch intended as a working sketch of an idea. Students will:

- propose a direction,
- build a functioning prototype, and
- identify what is promising and what requires further development.

This project emphasizes speculative programming, where structure emerges through iterative development.

Final Project — 40%
A performance-ready patch or system presented publicly at the end of the term. Possible formats

include a live instrument, installation concept, live-electronics setup with another performer, or a piece that relies on real-time processing. The system must be stable, documented, and prepared for public presentation.

Participation and Studio Engagement — 10%
Regular attendance, engagement in discussions, and participation in informal critiques and in-class work.

Tools & Materials

- **Software:** Max (Cycling '74)
- **Hardware:** Laptop and headphones; optional controllers, microphones, or sensors
- **Resources:** Max documentation, curated examples, short readings/listenings

Students from any background are welcome. The course centers on sound but supports work in performance, installation, and other media.