

Max/Msp/Jitter version 5 : a Project-Oriented Workshop with Jeremy Bernstein, Peter Castine, and John Dekron.
Participant level: Beginners and Intermediate users.

Monday 30 March- Friday April 3rd (Course Work)
11.00-19.00 daily with one hour lunch break
Final Presentation Friday April 3rd-5th

Location: NK / ElsenStr. 52 (2.Hof) Berlin, Germany
Telephone: +49(0)176 20626386

Course Participation fee: 325 €

email to: eNKa_NK@gmx.de

Please register early to ensure a place. Places are limited to 12

Essential:

Participants are required to bring their own laptops with Max/MSP/Jitter installed. The demo version can be downloaded from <<http://www.cycling74.com/downloads/max5>> and is fully functional in demo mode for up to 30 days after installation. Registered participants who require an extension of the demo period for the time up to and including the workshop can request this from the workshop organizer.

All participants **must** prepare for the workshop by completing Tutorials 1-14 included in the Max documentation, part of the download linked above. This will maximize the benefits of participating in this project-oriented workshop and is essential to achieve the goal of completing a working project in the course of one week. The final projects will be put on exhibition as an installation open for viewing on the weekend of April 3-5.

Max/MSP Topics:

We will rapidly move through the basics of the MaxMSP visual programming environment, answering any lingering questions of the participants (who are assumed to be mildly familiar with the system — they should have already worked through the first 14 tutorials) and getting everyone in the group up to speed with the essentials of patch construction, message passing, encapsulation, order of operations, data processing and so on.

Although MaxMSP is a large and complex set of building blocks, the construction of compact, understandable and (above all) useful patches is surprisingly simple. Within a couple of hours, participants should be comfortably creating basic patches and analyzing/modifying more complex existing patches to implement their ideas within the MaxMSP environment.

Jitter Topics:

Armed with the essentials, we'll move right into visual data processing using the Jitter extensions to MaxMSP. Topics to be explored include still and moving image playback and processing, use of external controllers or user interface devices to influence playback and processing behaviors, direct pixel access and transcoding (conversion of Jitter data to "normal" MaxMSP data and vice versa), among others.

Jitter extends the data vocabulary of MaxMSP (integer & floating-point numbers, symbols, lists) to include a 'matrix' type: a table of numbers (think Excel spreadsheet). These tables can have several layers/planes and multiple dimensions, making them an ideal carrier for (among other things) frame-based media such as video or OpenGL (3D) vertex and texture

data, text and audio analysis data. We'll be taking a survey of the most important and typical uses of Jitter matrices, particularly as they can be applied to the group's project.

Project Idea:

The core idea for the workshop project is to start from a small and simple video file—typically formatted 48x64 pixels, either monochrome (black-and-white) or 16 grays—and generate an acoustically interesting audio signal that will be used to encode and transmit the video to another computer. This will, in turn, use the video data to control simple everyday household electric devices.

Participants will need to bring at least one such household device with them. The device will be "operated" simply by opening and closing the power supply. It must be simple enough that its power switch can be left in the 'on' position, and electricity being turned on and off will control its operation. Possible devices: lamp, fan, portable electric grill or toaster, hair dryer, vacuum cleaner, power drill, older television. However, any device that automatically resets itself when power is cut off (most CD/DVD players and modern TVs do this) will not be suitable.

The devices may make a noise, produce light or heat, or create a smell. We want to involve as many senses in the installation as possible. Be imaginative.

Schedule:

- Day 1- 2
 - Introduction to Max: objects, messages, and patch cords
 - Jitter: introduction to image processing.
- Days 2 and 3
 - Building audio based protocol to transmit image based information
- Days 4 and 5
 - Attaching physical devices to the system
 - We will investigate different interfaces that enable Max to communicate with the outside world. This may include the Arduino board, the Teabox, DMX interfaces, and/or serial boards with attached relays. The devices that can be addressed may be of any kind: hair dryers, vacuum cleaners, vents, lamps, toasters... whatever you find in the kitchen or basement. The more noise, heat, light, dust, fog they make the better they are. The goal will be to orchestrate these devices so that they form some sort of composition—or at least that it is fun to watch.

Tools:

-[Arduino Board](#)

-DMX Interface

-DMX Dimmers

-USB to Serial Board with Relays

Workshop leaders:

Jeremy Bernstein makes temporal media, comprising mostly sound, video and code. In various forms—performances, presentations and screenings—his work has been exhibited throughout Europe and North America. For several years, his work has been concerned, on a technical level, with "reanimation"—assembling temporal structures out of static or

frozen fragments of "personally associative" materials—home movies, snapshots, field recordings, old records, etc. Increasingly, he has been exploring sound and image as "influenceable media" -- semi-fluid forms capable of withstanding structural modulation -- through the use of custom computer software. He has composed numerous music and sound scores for theater and dance productions, and constructed several interactive video and sound environments for theater and other live performance events.

He is also a developer of MaxMSP (Cycling '74, 1988 - present) and co-author of the Jitter multidimensional data processing environment (Cycling '74, 2002 - present).

<http://www.bootsquad.com/>

Peter Castine is a composer and sonic artist who has been using Max since 1990. He has produced electroacoustic music at the "Elektronisches Studio" at the Technical University of Berlin, the "Studio for Sonic Arts" at the University of the Arts in Berlin, and the Bergen (Norway) Center for Electronic Arts (bek). His electroacoustic and instrumental music has been performed in Europe, Asia, and North America. He is increasingly working with sonic and multimedia installation art, particularly the realiTV project where he works with John Dekron, and has shown work at Transmediale, ICMC, and other festivals. He has taught courses in Max/MSP at Darmstadt, Warsaw, and Berlin.

Castine is also known as a developer of external objects for Max/MSP/Litter, with projects including the Litter Power Package, the Litter Bundle for Jitter (LBJ), and the ICE Sequencing tools with (with DSPAudio). He has also developed code for the BIX media façade and the iPhone application FutureSound.

<http://www.bek.no/~pcastine/>

John Dekron is a media artist, performer and developer of customized video applications for media environments. His work and long-term media experience cover a broad spectrum: from media for architecture and video installations to custom setups and individual programming for clubs, artists and visual concepts for internationally operating companies and high-profile events. Past projects include performances in Germany, Europe and South America, programming of the BIX media facade in Graz, Austria, the SPOTS media facade in Berlin, works for musician Gonzales, the artist Carsten Nicolai plus multi-screen software development for Formula One side events in the Middle East. John Dekron is programmer and developer of the Softmix HD Video-Mixing and ES-X crossplatform VJ performance software.

He has given lectures and workshops on realtime video processing and performing at Merz Akademie, Stuttgart and the HBG, Leipzig, and at other venues.

<http://www.thismedia.com>

http://www.thisserver.de/john_dekron.php