Creating Collaborative Telematic Performances with Collab-Hub

ICMC 2022

Collab-Hub

Nick Hwang - University of Wisconsin - Whitewater

Anthony T. Marasco - University of Texas Rio Grande Valley

Eric Sheffield - Miami University

Collab-Hub Website

https://www.collab-hub.io/

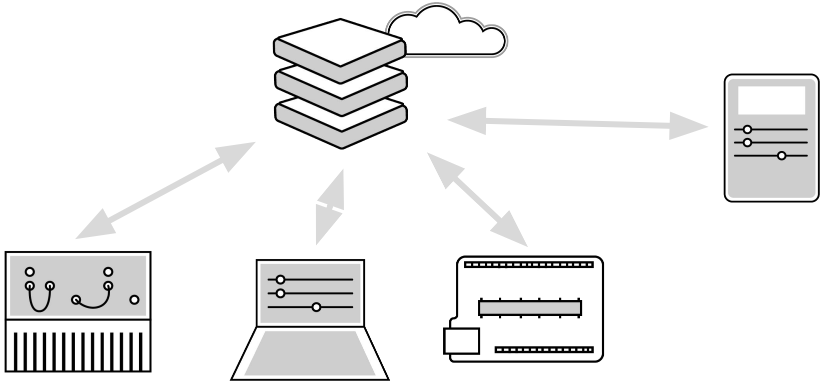
Find Links to download the Max Client and to join our Discord channel

General Workshop Outline

1. Introduction to Collab-Hub and the Collab-Hub Max Client Modules
2. Sending and Receiving Control and Event Data
3. Mapping Strategies and Collaboration Modalities
4. Understanding Push/Publish Distribution Types and Room Organization
5. Other Platforms
6. Max Demo Instruments

What is Collab-Hub?

- A server-based messaging tool for sharing data.
- Uses the Node.JS implementation and a remote server.
- Clients also available or in development for PD, Arduino, Web, Monome Norns, and more.



What ISN’T Collab-Hub?

It is not an audio-sharing tool.

It is not a video-sharing tool.

It’s not an instrument / set of instruments.

WHY is Collab-Hub?

Ease of Connectivity

Multi-platform and flexible

We built it for past us!

Collab-Hub in Use

SHP of THSEUS

- Uses Collab-Hub to distribute controls, advance score, and manage cues

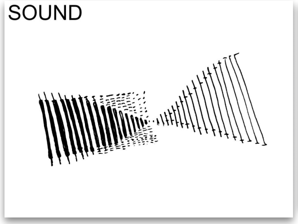
Rhumbline.io

- Collaborative installation with a web-based interface

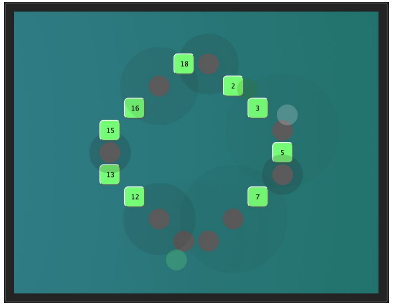
V.erses

- Interactive installation NYC

SHP of THSEUS score page & screenshot



Rhumbline.io



Workshop Material

Demo Files:

https://github.com/Collab-Hub-io/ICMC2022-CH-Workshop

Max Package: Collab-Hub

Web Interface:

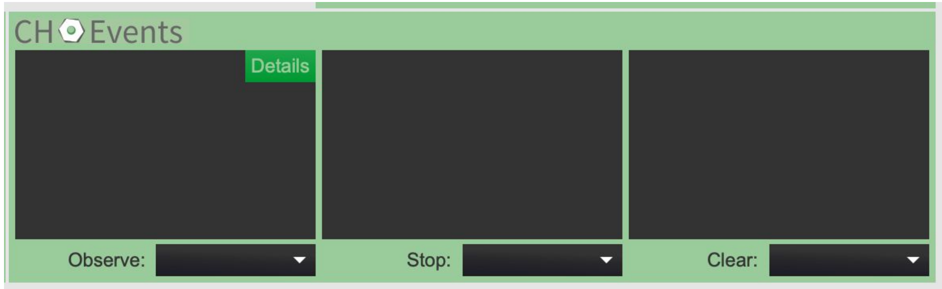
https://ch-server.herokuapp.com/

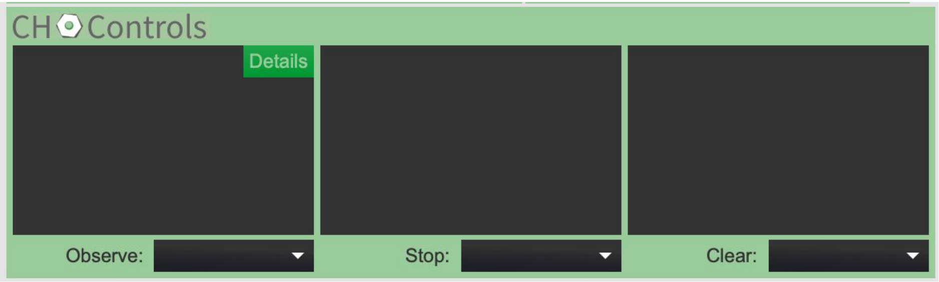
Sending and Receiving

Message Types

Collab-Hub messages are designated as either Control or Event messages

Used to indicate the intended mapping/use of the message and any corresponding data





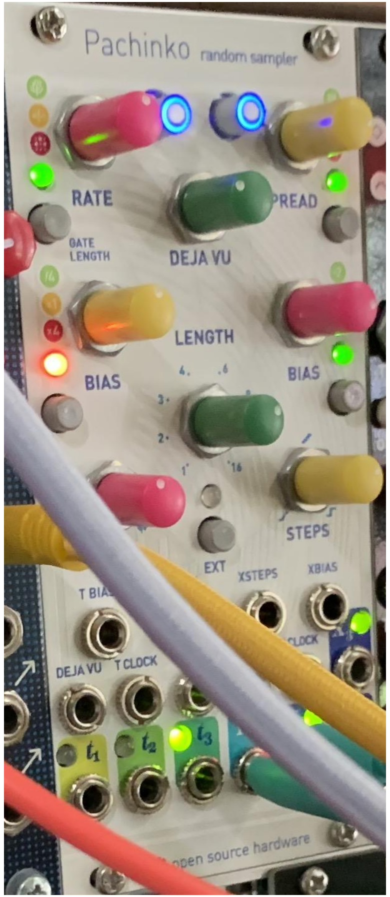
Control Messages

Control messages carry both a header and a value(s)

Header is a label showing intended purpose

Value can be an int, a flonum, or a symbol

A single Control message can also carry a list of mixed data types



Event Messages

Event messages only carry a header

Produces a bang, no value attached to message

A tap, a cue, a “go”

https://commons.wikimedia.org/w/index.php?search=pointing+finger&title=Special:MediaSearch&go=Go&type=image

Practice Sending and Receiving Messages

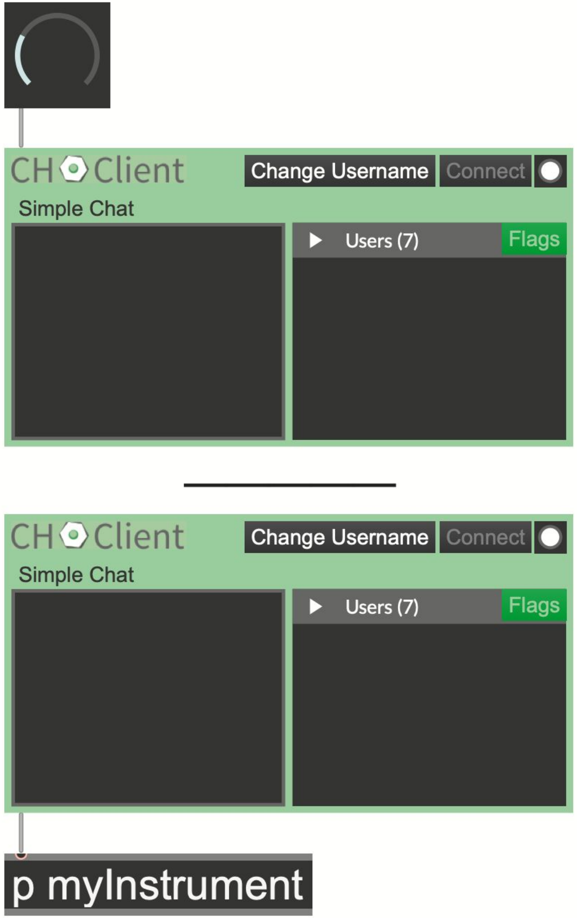
Event messages only carry a header

Mapping Strategies

Mapping Strategies

One-to-One

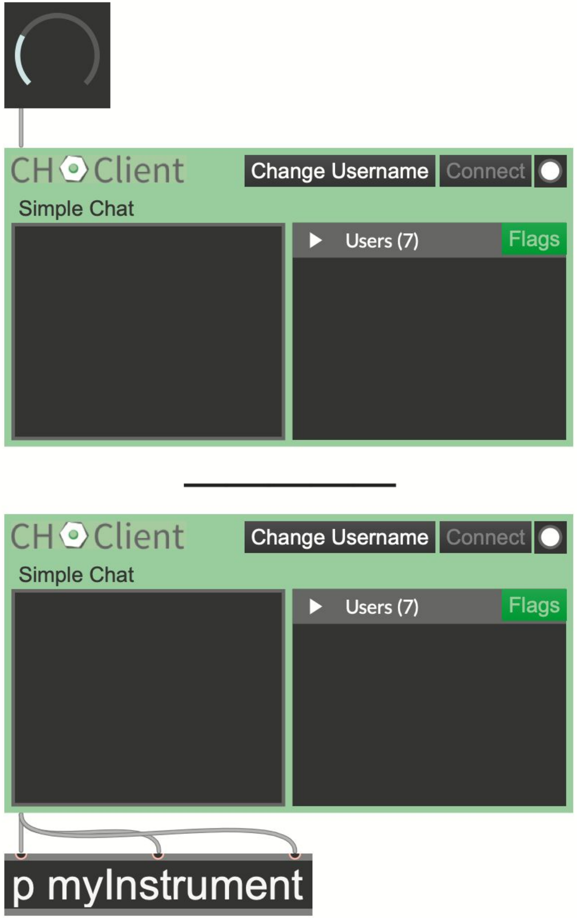
Connect incoming control directly to one parameter



Mapping Strategies

One-to-Many

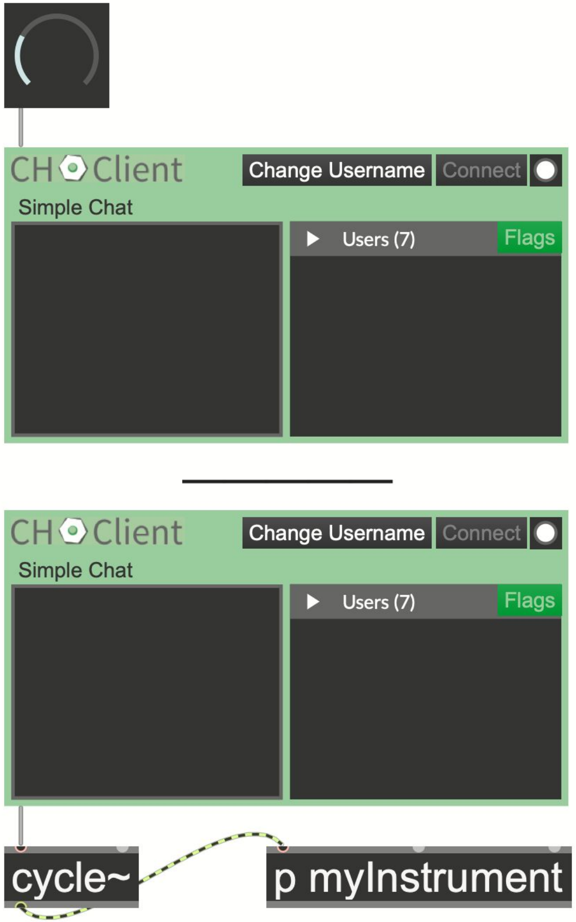
Connect incoming control directly to multiple parameters



Mapping Strategies

Driving LFOs

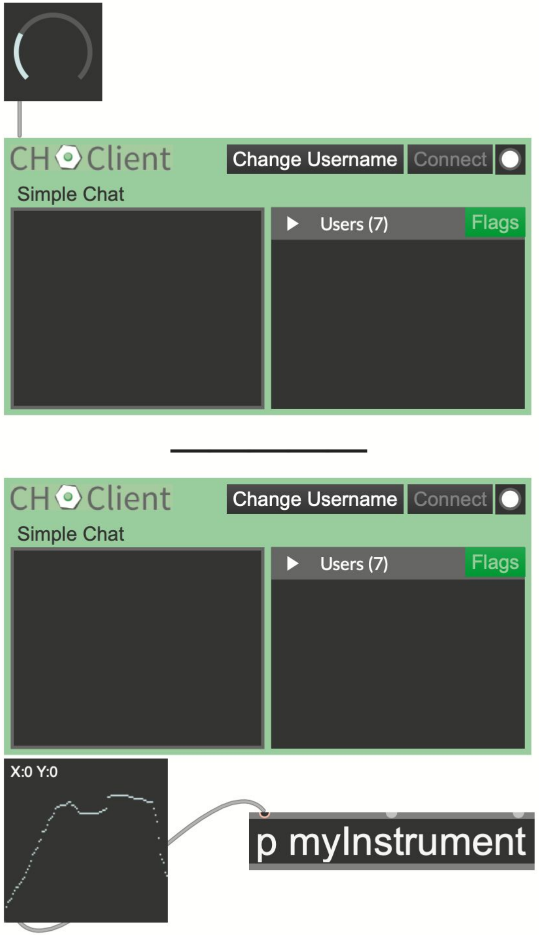
Connect incoming controls to LFO frequency inputs



Mapping Strategies

Using Lookup Tables

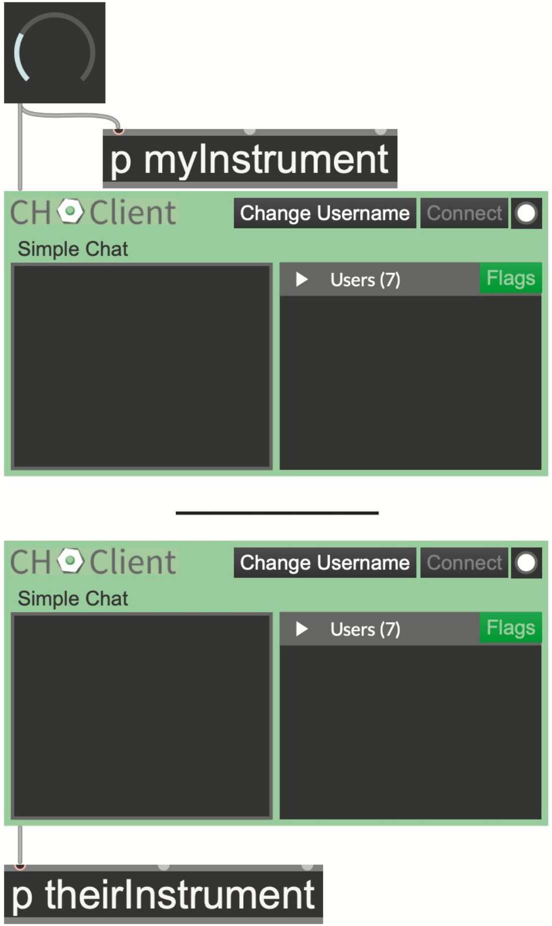
Connect incoming controls to a lookup table to modify the values



Mapping Strategies

Double Duty

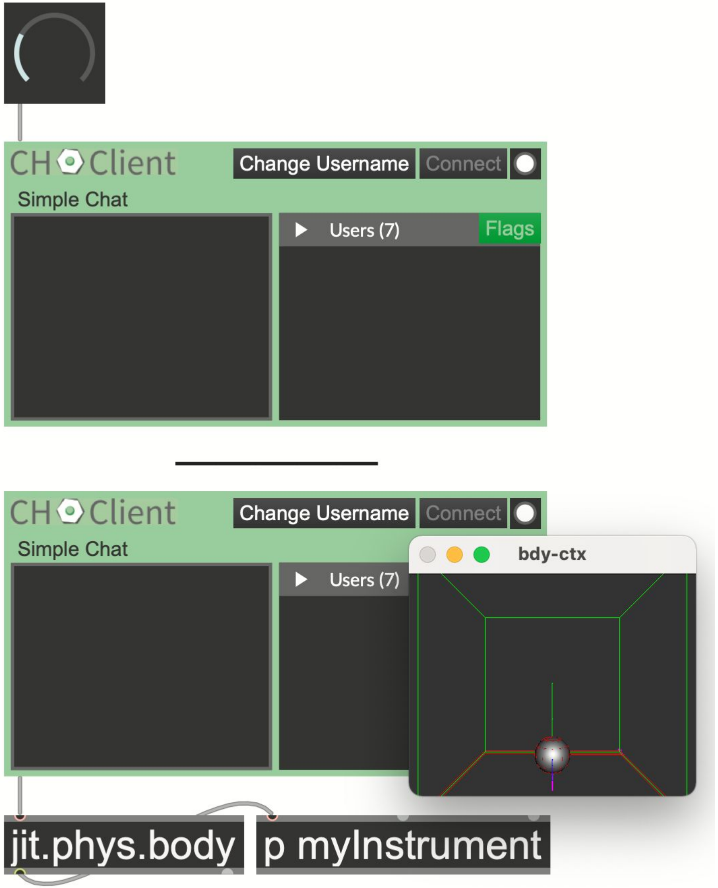
Send one of your own active instrument controls to other users



Mapping Strategies

Get Physical

Connect incoming controls to physical models for unpredictable or dynamic behaviors



Practice

Practice changing mapped parameters, scaling behaviors, and/or incorporating Collab-Hub mappings into your own instruments.

Collaboration Modalities

Some thoughts on agency

Thinking about how agency is distributed among players…

- Mutual agency
- Receiver asserts more agency
- Sender asserts more agency

Mutual agency

A direct and transparent implementation of shared control.

Sender: “Hey, I’m going to send you values 0-127 with the control header ‘from1’.”

Receiver: “Great, I’m going to connect that to my pitch shifter and scale values so that 0 is no pitch shift, 127 is an octave up.”

Receiver asserts more agency

Outcomes are transparent to the receiver, opaque to the sender.

Sender: “Hey, I’m going to send you values 0-127 with the control header ‘from1’.”

Receiver: “Great, I’m going to connect that to various parameters throughout the performance.”

Sender asserts more agency

Sender has more potential to intervene in receiver’s intentions.

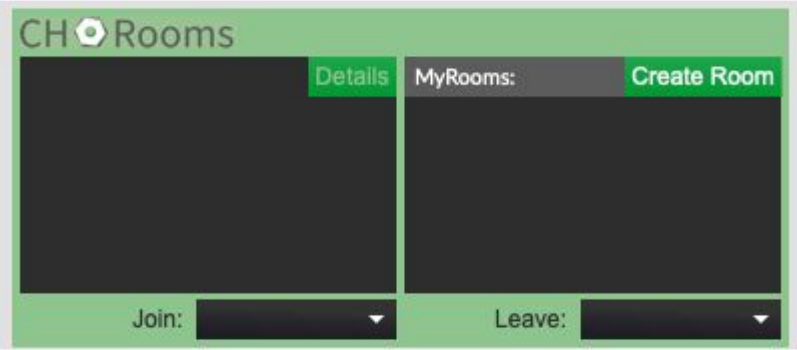
Receiver: “Hey, I have 4 parameters open to receiving control data.”

Sender: “Great, I’m going to bounce around and change different ones throughout the performance, regardless of your wishes!”

Distributing / Routing

Rooms

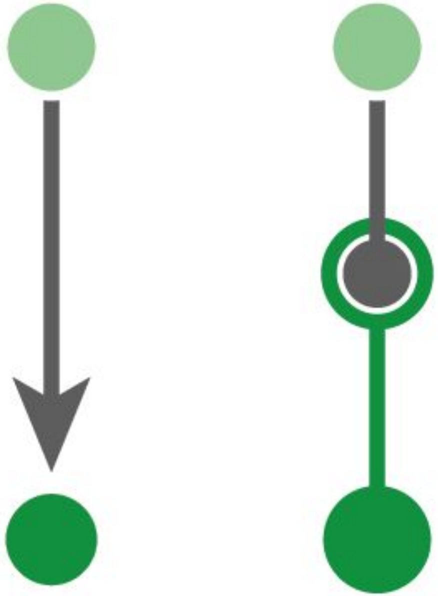
Users can create and join any number of rooms. Rooms can be a way to send/receive more than one user.



Push or Publish

Collab-Hub has two routing methods.

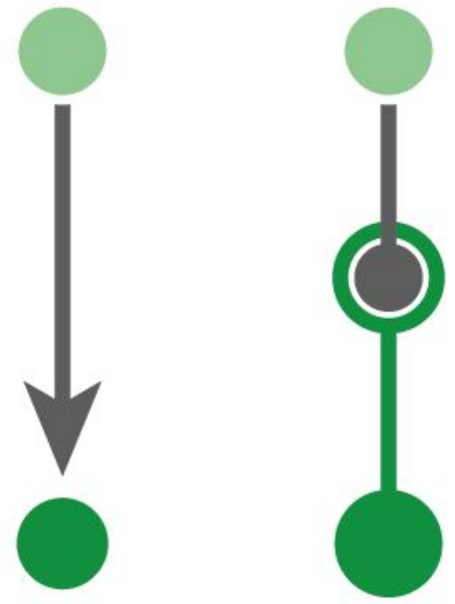
Push and Publish.



Push or Publish

All messages use a routing keyword of either ‘push’ or ‘publish’.

Depending on your usage, one should work for you.



Push

Send data to a destination.

Destinations:

- ‘all’
- <room name>
- <user name>

Syntax: push <destination> <header> <value> | Example: push all slider1 127

Push or Publish

Publish

Share data that can be observed.

Observers see available data, may that observe.

Publish

Data is made available to a destination.

Destinations:

- ‘all’
- <room name>
- <user name>

Publish

Send data to a destination.

Destinations:

- ‘all’
- <room name>
- <user name>

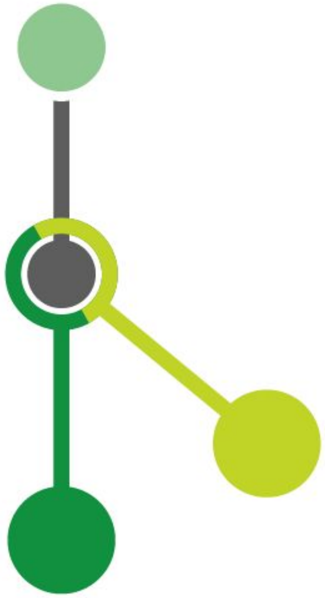
Syntax: publish <destination> <header> <value>

Example: publish all slider1 127

Publish

Observers can see available controls or events. They can choose to observe that data. Any future control value changes or events occurrences will be routed to the observer.

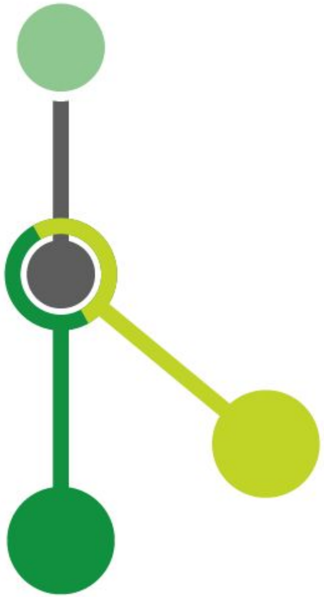
Any number of users can observe publish data.



Publish

Published Controls and Events are unique to the namespace.

At the moment, Once a user leaves the server, their published headers are released.



Other Platforms

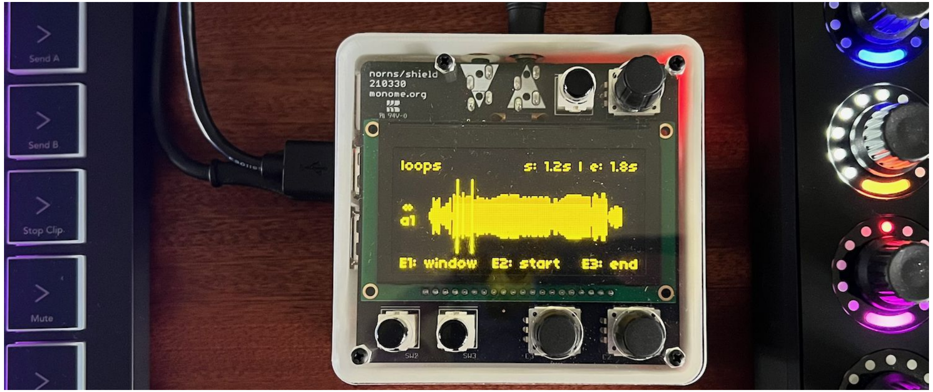
p5.js

Send and receive Collab-Hub data from your p5 sketches using the p5.CollabHub library.

Monome Norns

The Monome Norns is a tabletop CMI (running on a Raspberry Pi), popular in the boutique synthesizer scene

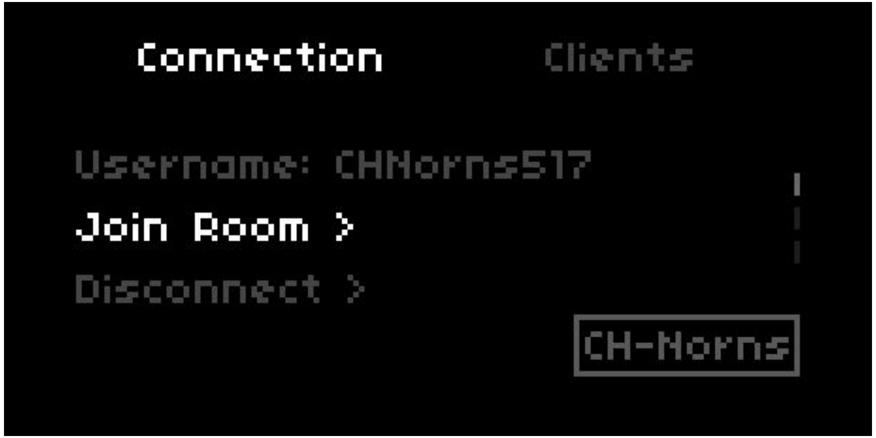
OS makes digital instrument “script” collection and param presets easy to manage



CH-Norns

CH-Norns is a “mod” script (runs in the background)

Passes controls/events over to scripts through OSC

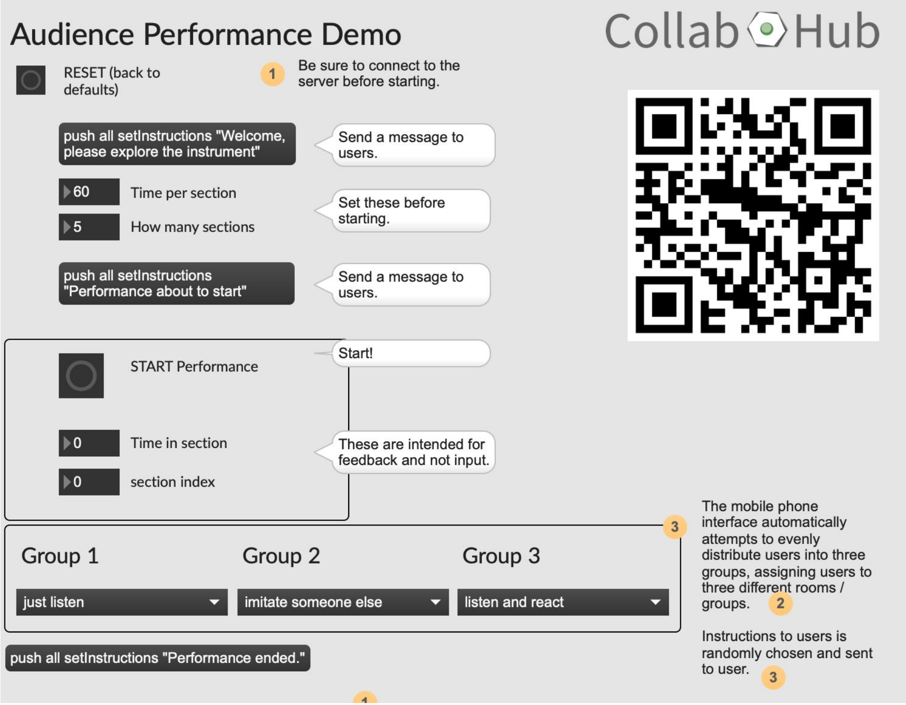


Username: CHNorns517

Tone + P5 + Max + Mobile

———

Demo of combination of Collab-Hub, P5, Max, Tone.JS, and mobile phones.



Audience Performance Demo

Collab Hub

RESET (back to defaults)

1 Be sure to connect to the server before starting.

push all setInstructions "Welcome, please explore the instrument"

Send a message to users.

60 Time per section

5 How many sections

Set these before starting.

push all setInstructions "Performance about to start"

Send a message to users.

START Performance

Start!

0 Time in section

0 section index

These are intended for feedback and not input.

Group 1	Group 2	Group 3
just listen	imitate someone else	listen and react

3 The mobile phone interface automatically attempts to evenly distribute users into three groups, assigning users to three different rooms / groups. 2

Instructions to users is randomly chosen and sent to user. 3

push all setInstructions "Performance ended."

Questions?

Nick - hwangn@uww.edu
Anthony - anthony.marasco@utrgv.edu
Eric - sheffie@miamioh.edu