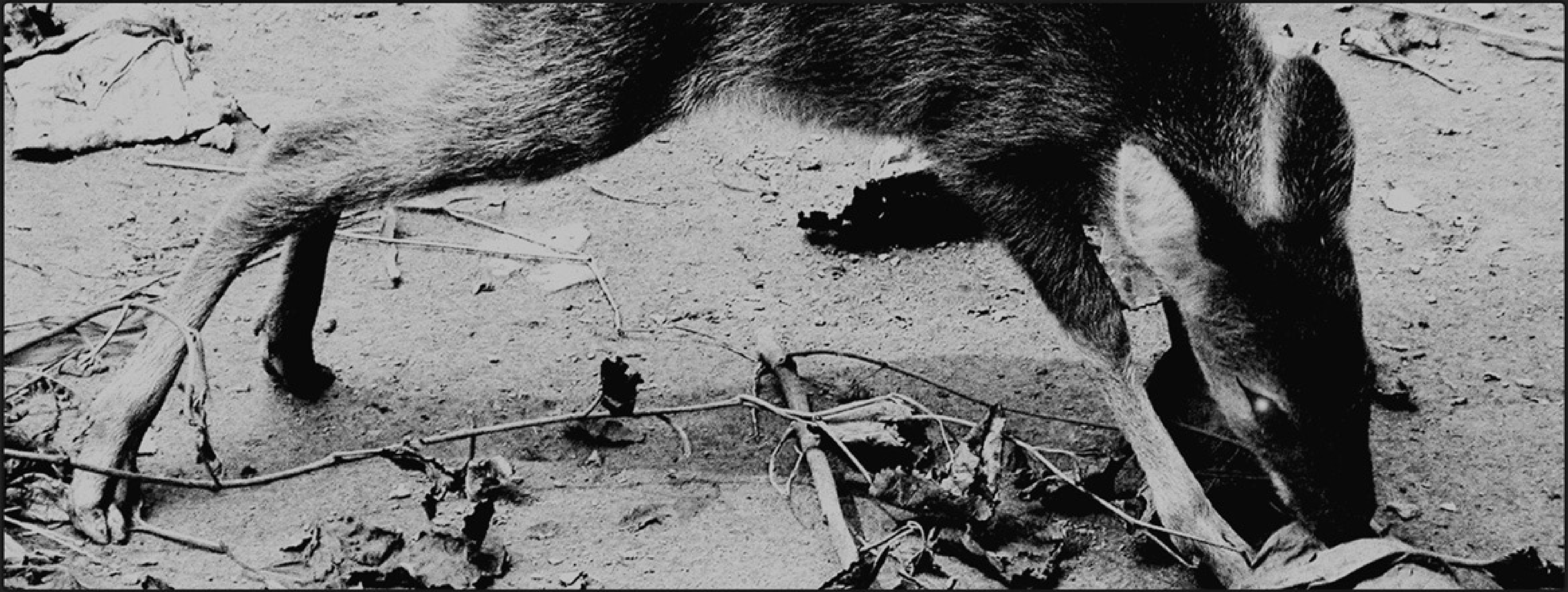
HYBRID LIVE CODING

Modular Application for interactive design development&performance&teaching

GAmuza v.03



WHAT is GAmuza

GAmuza is an Hybrid Live Coding/Modular application, for interactive design developing, live audiovisual performance and generative art teaching.
GAmuza is free software and is distributed under the terms of the MIT License (see page 17).

• Live Coding

The main core of GAmuza is a live coding environment for rapid prototyping, playing and learning.
Based on Lua embeddable scripting language, extends the entire OpenFrameworks API (at this moment v.007) with different input/output setting modules easy configurable within a GUI.

• Modular System

GAmuza works with various GUI modules. Every active module will be available, with all his output variables and functions, within the GAmuza live coding environment. For module list and detailed info see Using GAmuza section (page 10).

• Development>>Performance>> Teaching

GAmuza is a software for learning creative programming in a easy way, designed to cover the typical needs of teaching/presenting a work, realize an audiovisual live performance and rapid prototyping for developing new ideas.
Gamuza is easy as a scripting language, with coding extremely simplified through the modules, and with the blasting power of C++.

WHY GAmuza

The word ‘gamuza‘ in spanish means ‘shammy‘ or ‘chamois‘, that is an extremely agile goat antelope (Rupicapra rupicapra) of mountainous regions of Europe, having upright horns with backward-hooked tips.

The idea was to think in ‘extremely agile‘ concept, and realize a software for making the electronic art studying/understanding/developing more agile, jumping over all the typical technical issues and coding hangovers that always appears, especially in environments like fine arts or creative design, where the technical background, and everything related with computer science or electronics, use to be quiet low (at least here in europe).

This don’t means ‘make it really easy for stupid people‘; GAmuza is designed to simplify, but NOT to hide concepts or structures.

Simplify, to let the people lose fear at something that rumors said that it’s too complicated.

NOT hiding concepts or structures, because understanding the tools that we use, actually everyday, is the first step for don’t be a slave of technolgy.

- **Technology to the people. NOT otherwise**

  All of the sudden, everyone is publishing is personal ideas, interests, conversations, likes, opinions, photos, videos everywhere, and from everywhere.

  Everyone is online, form his/her laptop and last generation smartphone, and in 5 minutes you’ll be the great webmaster of all your various virtual lifes, constantly connected with all your virtual friends, creating personal fictions without the knowledge of technology you’re using; and this means a lot of things, one is ‘People to technology‘, or in other words ‘People to who prepare/sell technology for them‘.
  GAmuza is just a software, but created with one concept in mind: ‘Technology to the people, not otherwise‘.

Installing GAmuza

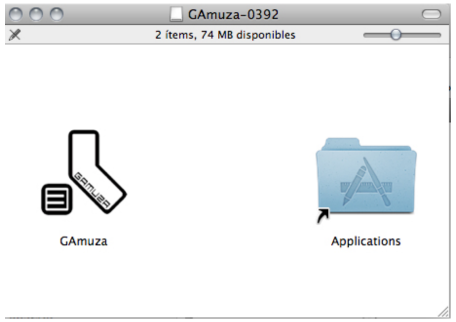
Download the last version of GAmuza (actually v.03) at the webpage:
http://www.gamuza.cc

In the download section, you'll see something like

GAmuza-xxxx.dmg
or
GAmuza-xxxx-Linux.tar.gz

where xxxx is the version you've downloaded

choose your OS (Linux or OSX), download and extract the content of the file; you'll have a folder with the same name of the file you download, everything you need is this folder with his content.
Copy this folder wherever you want in your computer.



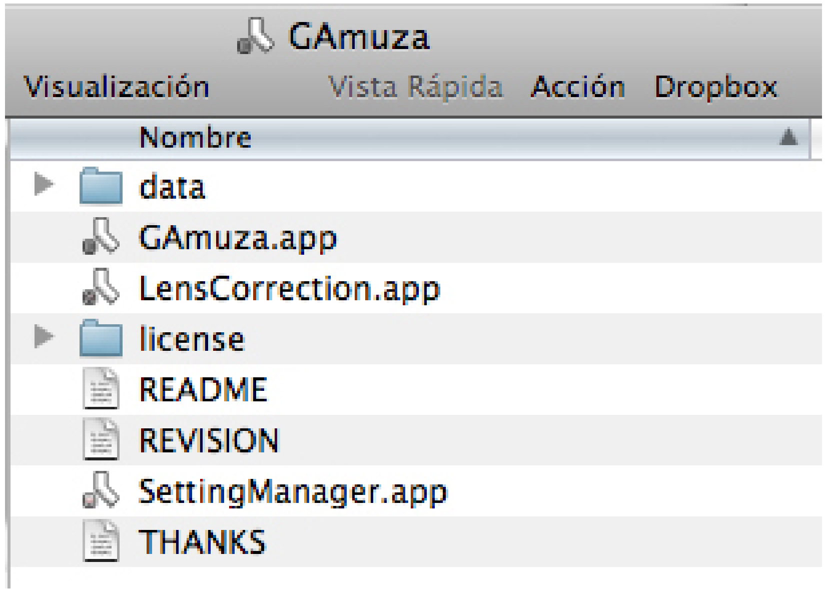
- **OS Support**

GAmuza software is actually available for -nix only systems (Linux and MAC OSX).

The source code of GAmuza is available online in the project page to download it and port it to other OS.
You can grab the last version code here:
https://github.com/d3cod3/GAmuza

• OSX

Inside GAmuza folder you'll find this files:



GAmuza

Visualización Vista Rápida Acción Dropbox

Nombre
data
GAmuza.app
LensCorrection.app
license
README
REVISION
SettingManager.app
THANKS

The SettingManager.app will help you to configure the modules and devices [see page 8]

LensCorrection.app helps you to correct lens aberrations on your video devices. [see page 11]

GAmuza.app, Just double click to launch the aplication [see page 12]

• LINUX

Download this file from GAmuza website:

```
Gamuza-xxxx-Linux.tar.gz
```

Save it wherever you want, open a terminal window, and extract it:

```
tar xvfz GAmuza-xxxx-Linux.tar.gz
```

[where xxxx is the release number you've downloaded]

This will create a folder named GAmuza, then change to that directory:

```
cd GAmuza
```

Before running GAmuza software is recommended to open the manual [located inside manual/ folder] and run the settingManager application [see page 8]:

```
./settingManager
```

or open nautilus file manager and double click on settingManager.

You can now run GAmuza from terminal window typing:

```
./GAmuza
```

or open nautilus file manager and double click on GAmuza.

- **Minimal Hardware**

Due to the modular nature of GAmuza software, there is not a fixed minimal hardware configuration.

The software core use more or less 150mB of RAM memory.

The rendering part of the software auto detect your graphic cards capabilities and switch automatically between shader-mode and no-shader-mode.

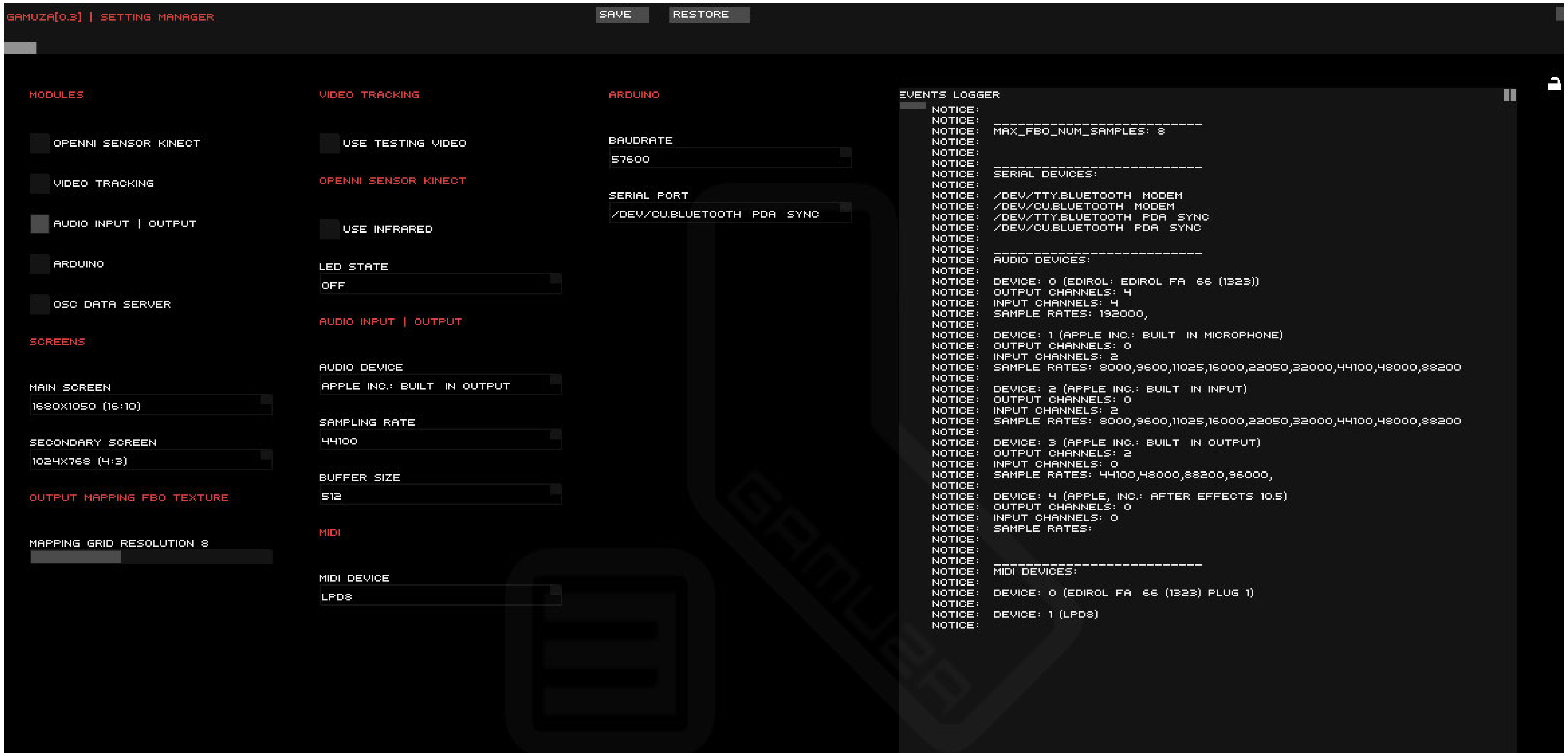
Depending of what module are you using and how many at the same time (for example, it's not the same using gamuza with one webcam module and the osc module, than using it with the sensor Kinect Module + 4 audio input channel modules + 2 webcam modules + the arduino module, obviously), the CPU load will be different.

Anyway, minimum requirement is:
2 Gb RAM
first generation Dual Core CPU

Minimal screen resolution (main screen):
1366x768 pixels

Configuring GAmuza:

SettingManager.app



MODULES

- ☐ OPENNI SENSOR KINECT
- ☐ VIDEO TRACKING
- ☑ AUDIO INPUT | OUTPUT
- ☐ ARDUINO
- ☐ OSC DATA SERVER

SCREENS

MAIN SCREEN: 1680X1050 (16:10)

SECONDARY SCREEN: 1024X768 (4:3)

OUTPUT MAPPING FBO TEXTURE

MAPPING GRID RESOLUTION 8

VIDEO TRACKING

- ☐ USE TESTING VIDEO

OPENNI SENSOR KINECT

- ☐ USE INFRARED

LED STATE: OFF

AUDIO INPUT | OUTPUT

AUDIO DEVICE: APPLE INC.: BUILT IN OUTPUT

SAMPLING RATE: 44100

BUFFER SIZE: 512

MIDI

MIDI DEVICE: LPD8

ARDUINO

BAUDRATE: 57600

SERIAL PORT: /DEV/CU.BLUETOOTH PDA SYNC

EVENTS LOGGER

```
NOTICE:
NOTICE:  ____________________________
NOTICE:  MAX_FBO_NUM_SAMPLES: 8
NOTICE:
NOTICE:
NOTICE:  ____________________________
NOTICE:  SERIAL DEVICES:
NOTICE:
NOTICE:  /DEV/TTY.BLUETOOTH MODEM
NOTICE:  /DEV/CU.BLUETOOTH MODEM
NOTICE:  /DEV/TTY.BLUETOOTH PDA SYNC
NOTICE:  /DEV/CU.BLUETOOTH PDA SYNC
NOTICE:
NOTICE:  ____________________________
NOTICE:  AUDIO DEVICES:
NOTICE:
NOTICE:  DEVICE: 0 (EDIROL: EDIROL FA 66 (1323))
NOTICE:  OUTPUT CHANNELS: 4
NOTICE:  INPUT CHANNELS: 4
NOTICE:  SAMPLE RATES: 192000,
NOTICE:
NOTICE:  DEVICE: 1 (APPLE INC.: BUILT IN MICROPHONE)
NOTICE:  OUTPUT CHANNELS: 0
NOTICE:  INPUT CHANNELS: 2
NOTICE:  SAMPLE RATES: 8000,9600,11025,16000,22050,32000,44100,48000,88200
NOTICE:
NOTICE:  DEVICE: 2 (APPLE INC.: BUILT IN INPUT)
NOTICE:  OUTPUT CHANNELS: 0
NOTICE:  INPUT CHANNELS: 2
NOTICE:  SAMPLE RATES: 8000,9600,11025,16000,22050,32000,44100,48000,88200
NOTICE:
NOTICE:  DEVICE: 3 (APPLE INC.: BUILT IN OUTPUT)
NOTICE:  OUTPUT CHANNELS: 2
NOTICE:  INPUT CHANNELS: 0
NOTICE:  SAMPLE RATES: 44100,48000,88200,96000,
NOTICE:
NOTICE:  DEVICE: 4 (APPLE, INC.: AFTER EFFECTS 10.5)
NOTICE:  OUTPUT CHANNELS: 0
NOTICE:  INPUT CHANNELS: 0
NOTICE:  SAMPLE RATES:
NOTICE:
NOTICE:
NOTICE:  ____________________________
NOTICE:  MIDI DEVICES:
NOTICE:
NOTICE:  DEVICE: 0 (EDIROL FA 66 (1323) PLUG 1)
NOTICE:
NOTICE:  DEVICE: 1 (LPD8)
NOTICE:
```

GAmuza comes with a configuring application for set up devices and modules you want to use: "SettingManager.app" for mac OSX or "SettingManager" for Linux x86, located in the main folder. [page 6]

Just open it and select the modules that you want to enable, and disable the unnecessary.

The moules options are:

OPEN SENSOR KINECT
VIDEO TRACKING
AUDIO INPUT ANALISYS
ARDUINO
OSC DATA SERVER

Next section is SCREENS:

MAIN SCREEN: Choose resolution
SECONDARY SCREEN: Choose resolution accordingly with your monitor(s) or projector(s).

OUTPUT MAPPING FBO TEXTURE

You can choose the warping grid definition, with a minimum of 1 (that means just 1 quad for deforming the texture) and a maximum of 20 (that means 400 quads, a lot).

If Video tracking module is selected: you can enable or disable TESTING VIDEO.

Just for testing, this setting is choosing to deactivate the webcams and use a video file instead, that means have the tracking module loaded, but working with a video file.

If Openni sensor Kinect module is selected:

Simple questions here, you can choose if use infrared vision (enable) or normal vision(disable) and if you want the front led of sensor Kinect ON or OFF, and if ON, choose color and if blink or not.

If Audio input|output module is selected: Choose the audio device, Sampling Rate and Buffer size.

Regarding the audio device ID, GAmuza software will give you a console print of all the technical data of devices on your system (audio, serial, midi), so look at the console output and check your audio device ID.

If you have one connected, choose the midi device

If Arduino module is selected: Choose BAUDRATE, and the SERIAL PORT.

Here we control the baudrate (57600 by default) and, most important, the serial device name,select the correct device port name here.

Save the configuration and GAmuza will open with the selected modules and devices.

REMEMBER, if you activate the openni sensor Kinect module (for example) and you don't have a sensor Kinect connected with your machine, GAmuza will probably CRASH. This is the same for the other modules, don't activate them if you don't have proper devices connected.

Advanced Settings

To configure advanced settings, you can access to the configuration file: GAmuza/data/settings/gamuzaSettings.xml

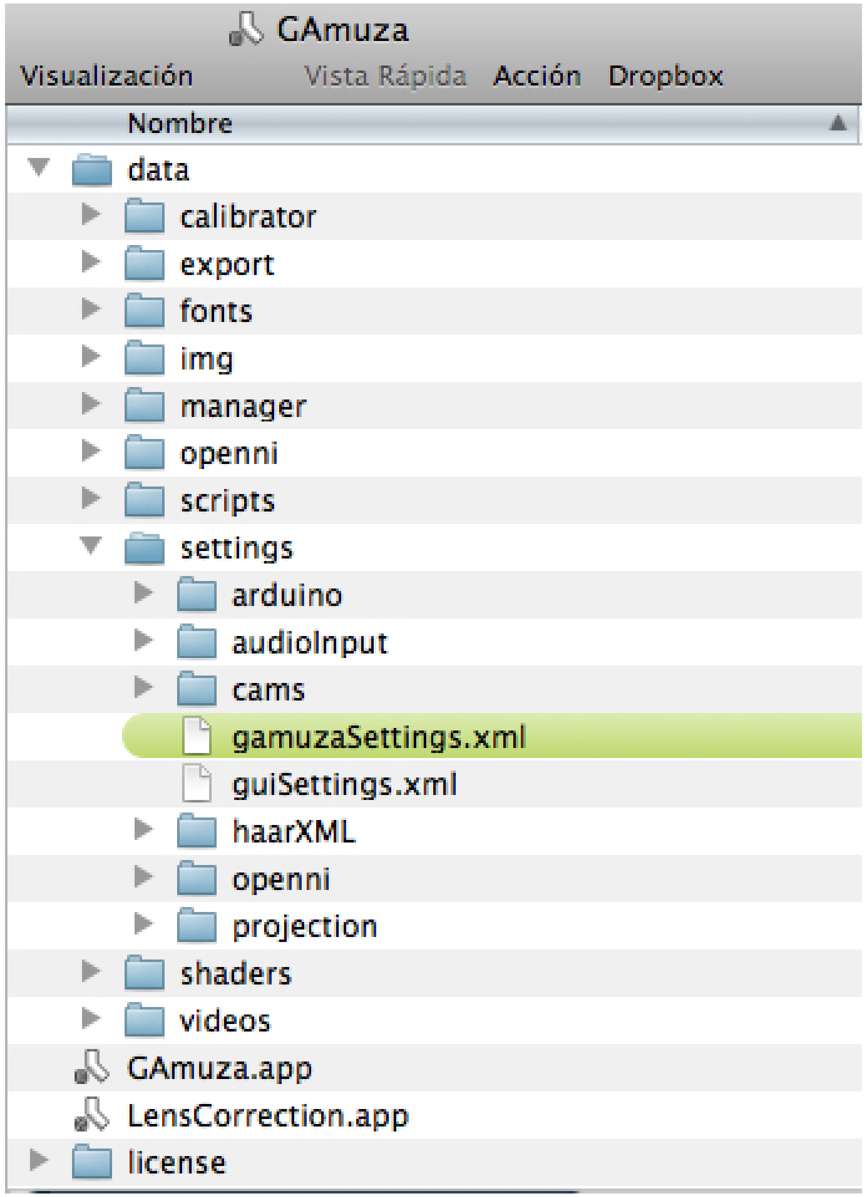
Just open it with a text editor or xml editor (better choice), and manually edit the file to change GAmuza settings.

> Use 1 if you want to activate the module, 0 otherwise.

For example, to the **Web Cam Tracking section**, in GAmuza the default resolution is 320x240, that for tracking algorithms and ´normal´ requirements, works really good and fast. Anyway, if your cameras support other resolutions and your machine is really a beast, then you can try to change this values.

The last setting of this seccion is choosing the Haar Finder file (if you're using this module and if you're computing this algorithm); all the Haar Finder files are located in GAmuza/data/settings/haarXML

To the **OSC module**, the OSC output bridge settings is just the IP number where to send data at what PORT.

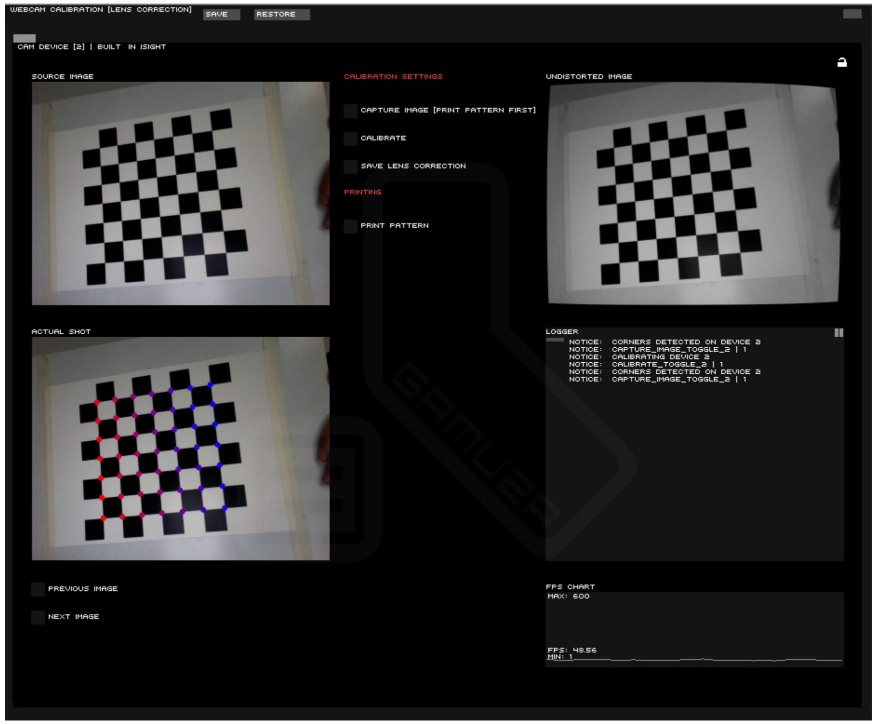


```
▼ data
    ▶ calibrator
    ▶ export
    ▶ fonts
    ▶ img
    ▶ manager
    ▶ openni
    ▶ scripts
    ▼ settings
        ▶ arduino
        ▶ audioInput
        ▶ cams
          gamuzaSettings.xml
          guiSettings.xml
        ▶ haarXML
        ▶ openni
        ▶ projection
    ▶ shaders
    ▶ videos
  GAmuza.app
  LensCorrection.app
▶ license
```

Lens Correction

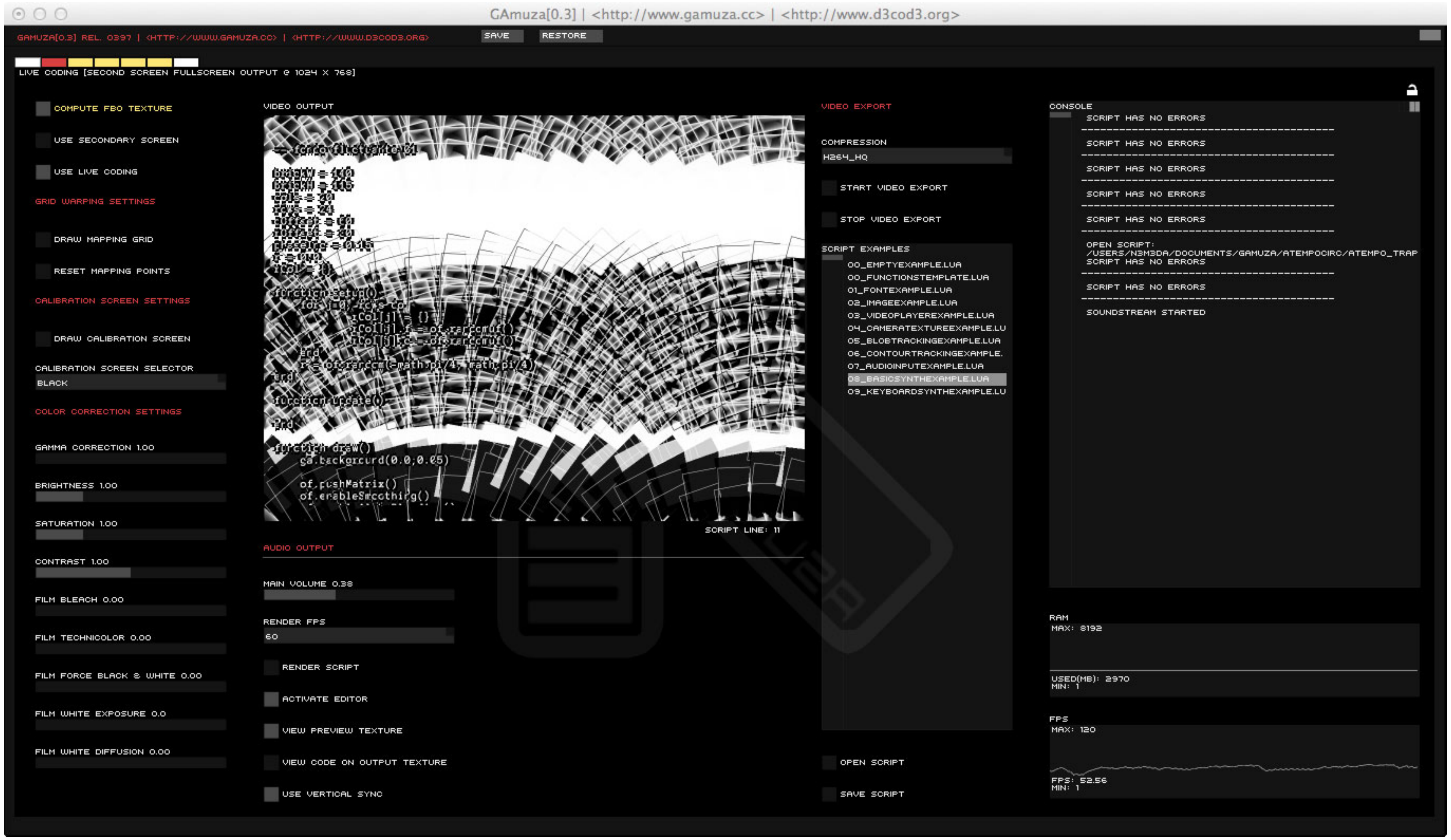
GAmuza has a small application to correct the aberrations of the video cameras lenses.

You can print de pattern from GAmuza. Then, capture this pattern image, calibrate it, and save the correction.



```
NOTICE:  CORNERS DETECTED ON DEVICE 2
NOTICE:  CAPTURE_IMAGE_TOGGLE_2 | 1
NOTICE:  CALIBRATING DEVICE 2
NOTICE:  CALIBRATE_TOGGLE_2 | 1
NOTICE:  CORNERS DETECTED ON DEVICE 2
NOTICE:  CAPTURE_IMAGE_TOGGLE_2 | 1
```

Using GAmuza



GAmuza[0.3] | <http://www.gamuza.cc> | <http://www.d3cod3.org>

GAMUZA[0.3] REL. 0397 | <HTTP://WWW.GAMUZA.CC> | <HTTP://WWW.D3COD3.ORG>

LIVE CODING [SECOND SCREEN FULLSCREEN OUTPUT @ 1024 X 768]

COMPUTE FBO TEXTURE
USE SECONDARY SCREEN
USE LIVE CODING

GRID WARPING SETTINGS
DRAW MAPPING GRID
RESET MAPPING POINTS

CALIBRATION SCREEN SETTINGS
DRAW CALIBRATION SCREEN
CALIBRATION SCREEN SELECTOR: BLACK

COLOR CORRECTION SETTINGS
GAMMA CORRECTION 1.00
BRIGHTNESS 1.00
SATURATION 1.00
CONTRAST 1.00
FILM BLEACH 0.00
FILM TECHNICOLOR 0.00
FILM FORCE BLACK & WHITE 0.00
FILM WHITE EXPOSURE 0.0
FILM WHITE DIFFUSION 0.00

VIDEO OUTPUT

```
-- [illegible]

brickW = 140
brickH = 115
cols = 20
rows = 24
xoffset = 60
roffset = 40
pressure = 0.15
r = 0.40
icol = {}

function setup()
    for i=0, rows do
        rCol[i] = {}
        rCol[i].r = of.randomuf()
        rCol[i].c = of.randomuf()
    end
    r = of.random(-math.pi/4, math.pi/4)
end

function update()

end

function draw()
    ga.background(0.0,0.05)

    of.pushMatrix()
    of.enableSmoothing()
```

SCRIPT LINE: 11

AUDIO OUTPUT
MAIN VOLUME 0.38
RENDER FPS: 60
RENDER SCRIPT
ACTIVATE EDITOR
VIEW PREVIEW TEXTURE
VIEW CODE ON OUTPUT TEXTURE
USE VERTICAL SYNC

VIDEO EXPORT
COMPRESSION: H264_HQ
START VIDEO EXPORT
STOP VIDEO EXPORT

SCRIPT EXAMPLES
00_EMPTYEXAMPLE.LUA
00_FUNCTIONSTEMPLATE.LUA
01_FONTEXAMPLE.LUA
02_IMAGEEXAMPLE.LUA
03_VIDEOPLAYEREXAMPLE.LUA
04_CAMERATEXTUREEXAMPLE.LU
05_BLOBTRACKINGEXAMPLE.LUA
06_CONTOURTRACKINGEXAMPLE.
07_AUDIOINPUTEXAMPLE.LUA
08_BASICSYNTHEXAMPLE.LUA
09_KEYBOARDSYNTHEXAMPLE.LU

OPEN SCRIPT
SAVE SCRIPT

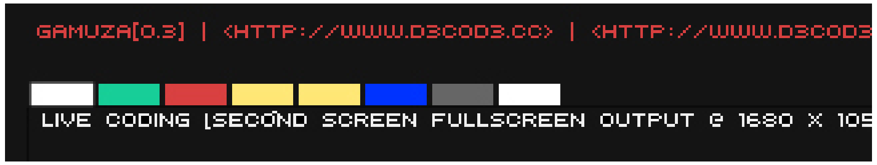
CONSOLE

```
SCRIPT HAS NO ERRORS
-----------------------------------------
SCRIPT HAS NO ERRORS
-----------------------------------------
SCRIPT HAS NO ERRORS
-----------------------------------------
SCRIPT HAS NO ERRORS
-----------------------------------------
SCRIPT HAS NO ERRORS
-----------------------------------------
OPEN SCRIPT:
/USERS/N3M3DA/DOCUMENTS/GAMUZA/ATEMPOCIRC/ATEMPO_TRAP
SCRIPT HAS NO ERRORS
-----------------------------------------
SCRIPT HAS NO ERRORS
-----------------------------------------
SOUNDSTREAM STARTED
```

RAM
MAX: 8192
USED(MB): 2970
MIN: 1

FPS
MAX: 120
FPS: 52.56
MIN: 1

GAmuza works with various GUI modules. When you open it, the main interface is the Live Coding one. In the upper left corner there are small tabs to access other modules.


GAMUZA[0.3] | <HTTP://WWW.D3COD3.CC> | <HTTP://WWW.D3COD3
LIVE CODING [SECOND SCREEN FULLSCREEN OUTPUT @ 1680 X 105

White:	Live coding
Green:	Openni Sensor Kinect
Red:	Tracking video
Yellow:	Audio input [You'll have as many tabs as inputs in the audio card]
Blue:	Arduino
Gray:	OSC sending server

Live Coding module

Toggle options to enable or disable:
COMPUTE FBO TEXTURE
USE SECONDARY SCREEN
USE LIVE CODING

Grid warping settings section:
DRAW MAPPING GRID
RESET MAPPING GRID

Calibration screen settings section:
DRAW CALIBRATION SCREEN SELECTOR:
Dropdown menu with options to:
Black, Cross, White, Test pattern color, Test pattern B&W; Webcam

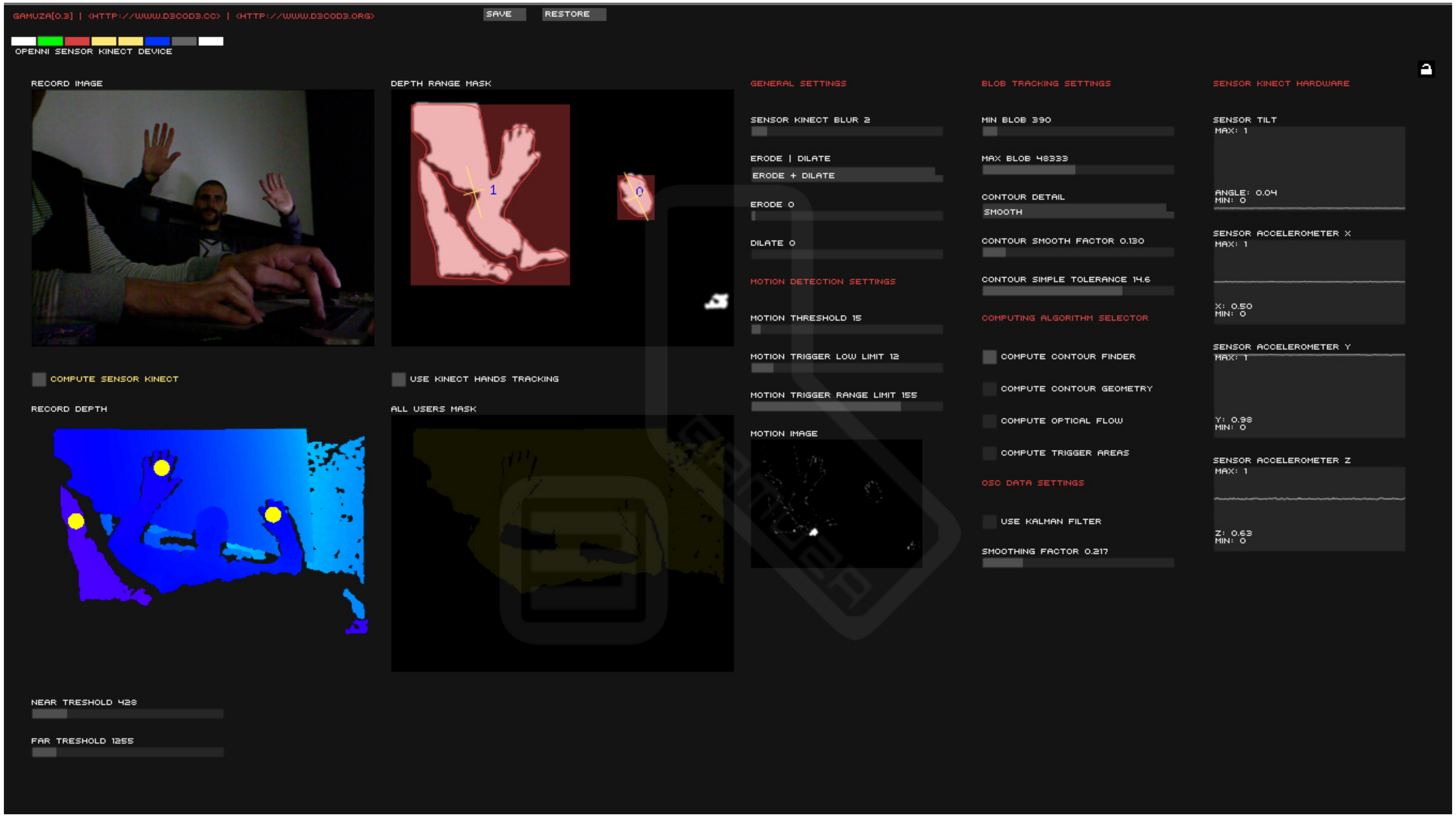
Color correction settings section.
Sliders to regulate the filters:
GAMMA CORRECTION
BRIGHTNESS
SATURATION
CONTRAST
FILM BLEACH
FILM TECHNICOLOR
FILM FORCE BLACK & WHITE
FILM WHITE EXPOSURE
FILM WHITE DIFFUSION

In the center is the console:
OUTPUT PROJECTION PREVIEW
[you can enable/disable it with the toggle COMPUTE FBO TEXTURE]

And, to the right:
File listed: SCRIPT EXAMPLES [from data/scripts/ folder]
[Double click to charge the script in the OUTPUT PROJECTION PREVIEW console]
Below, two buttons to open and save the scripts.

GAmuza's console

Sensor Kinect Module



GAMUZA[0.3] | <HTTP://WWW.D3COD3.CC> | <HTTP://WWW.D3COD3.ORG>

SAVE RESTORE

OPENNI SENSOR KINECT DEVICE

RECORD IMAGE

DEPTH RANGE MASK

COMPUTE SENSOR KINECT

USE KINECT HANDS TRACKING

RECORD DEPTH

ALL USERS MASK

NEAR TRESHOLD 428

FAR TRESHOLD 1255

GENERAL SETTINGS

SENSOR KINECT BLUR 2

ERODE | DILATE

ERODE + DILATE

ERODE 0

DILATE 0

MOTION DETECTION SETTINGS

MOTION THRESHOLD 15

MOTION TRIGGER LOW LIMIT 12

MOTION TRIGGER RANGE LIMIT 155

MOTION IMAGE

BLOB TRACKING SETTINGS

MIN BLOB 390

MAX BLOB 48333

CONTOUR DETAIL

SMOOTH

CONTOUR SMOOTH FACTOR 0.130

CONTOUR SIMPLE TOLERANCE 14.6

COMPUTING ALGORITHM SELECTOR

COMPUTE CONTOUR FINDER

COMPUTE CONTOUR GEOMETRY

COMPUTE OPTICAL FLOW

COMPUTE TRIGGER AREAS

OSC DATA SETTINGS

USE KALMAN FILTER

SMOOTHING FACTOR 0.217

SENSOR KINECT HARDWARE

SENSOR TILT
MAX: 1
ANGLE: 0.04
MIN: 0

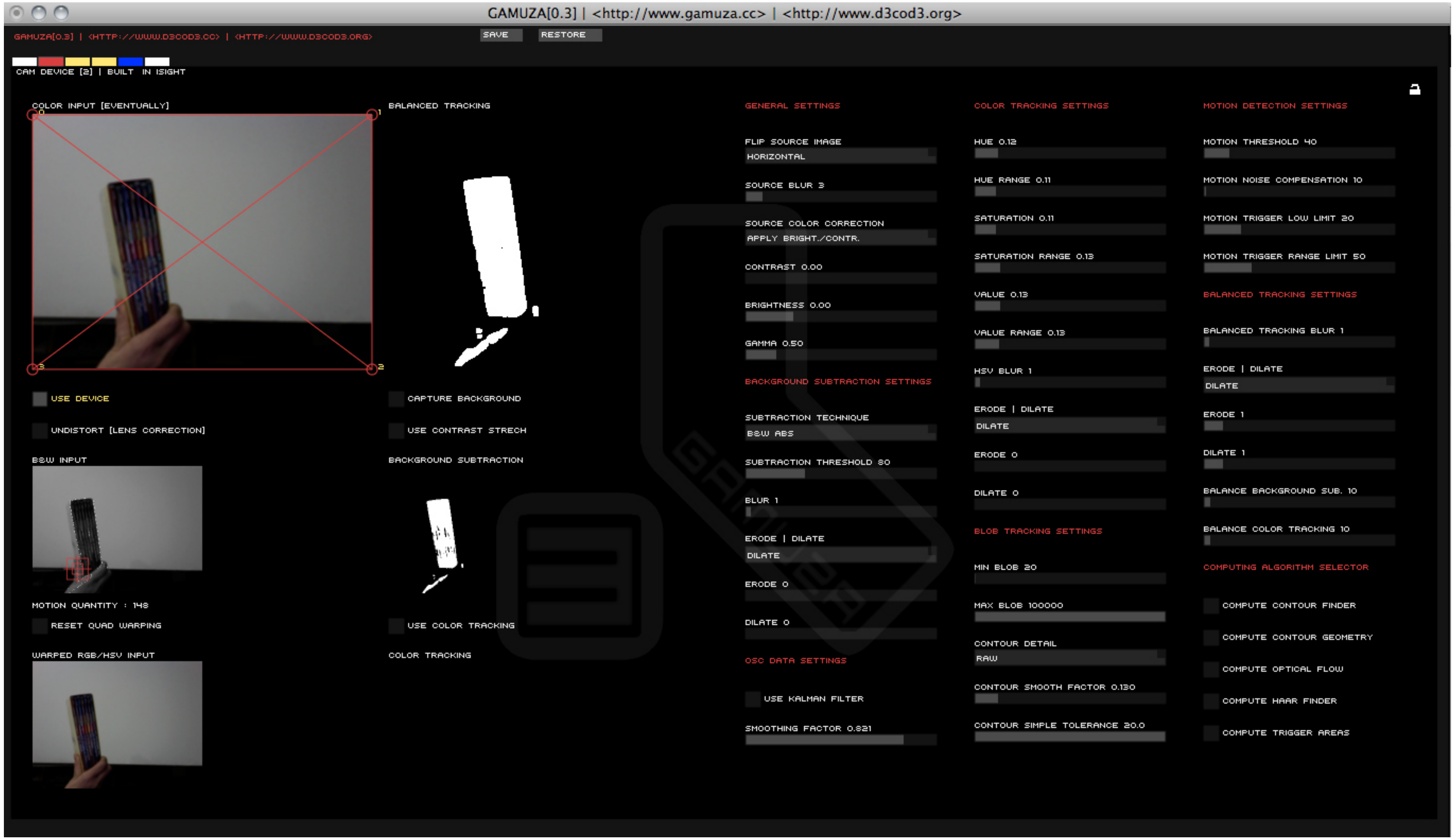
SENSOR ACCELEROMETER X
MAX: 1
X: 0.50
MIN: 0

SENSOR ACCELEROMETER Y
MAX: 1
Y: 0.98
MIN: 0

SENSOR ACCELEROMETER Z
MAX: 1
Z: 0.63
MIN: 0

Kinect module manual,soon.

Computer Vision Module(s)



GAMUZA[0.3] | <http://www.gamuza.cc> | <http://www.d3cod3.org>

GAMUZA[0.3] | <HTTP://WWW.D3COD3.CC> | <HTTP://WWW.D3COD3.ORG>

SAVE RESTORE

CAM DEVICE [2] | BUILT IN ISIGHT

COLOR INPUT [EVENTUALLY]

BALANCED TRACKING

USE DEVICE

CAPTURE BACKGROUND

UNDISTORT [LENS CORRECTION]

USE CONTRAST STRECH

B&W INPUT

BACKGROUND SUBTRACTION

MOTION QUANTITY : 148

RESET QUAD WARPING

USE COLOR TRACKING

WARPED RGB/HSV INPUT

COLOR TRACKING

GENERAL SETTINGS

FLIP SOURCE IMAGE
HORIZONTAL

SOURCE BLUR 3

SOURCE COLOR CORRECTION
APPLY BRIGHT./CONTR.

CONTRAST 0.00

BRIGHTNESS 0.00

GAMMA 0.50

BACKGROUND SUBTRACTION SETTINGS

SUBTRACTION TECHNIQUE
B&W ABS

SUBTRACTION THRESHOLD 80

BLUR 1

ERODE | DILATE
DILATE

ERODE 0

DILATE 0

OSC DATA SETTINGS

USE KALMAN FILTER

SMOOTHING FACTOR 0.821

COLOR TRACKING SETTINGS

HUE 0.12

HUE RANGE 0.11

SATURATION 0.11

SATURATION RANGE 0.13

VALUE 0.13

VALUE RANGE 0.13

HSV BLUR 1

ERODE | DILATE
DILATE

ERODE 0

DILATE 0

BLOB TRACKING SETTINGS

MIN BLOB 20

MAX BLOB 100000

CONTOUR DETAIL
RAW

CONTOUR SMOOTH FACTOR 0.130

CONTOUR SIMPLE TOLERANCE 20.0

MOTION DETECTION SETTINGS

MOTION THRESHOLD 40

MOTION NOISE COMPENSATION 10

MOTION TRIGGER LOW LIMIT 20

MOTION TRIGGER RANGE LIMIT 50

BALANCED TRACKING SETTINGS

BALANCED TRACKING BLUR 1

ERODE | DILATE
DILATE

ERODE 1

DILATE 1

BALANCE BACKGROUND SUB. 10

BALANCE COLOR TRACKING 10

COMPUTING ALGORITHM SELECTOR

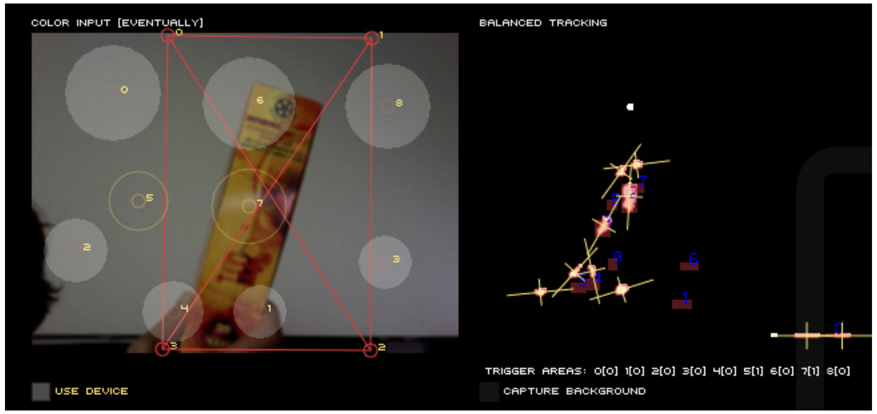
COMPUTE CONTOUR FINDER

COMPUTE CONTOUR GEOMETRY

COMPUTE OPTICAL FLOW

COMPUTE HAAR FINDER

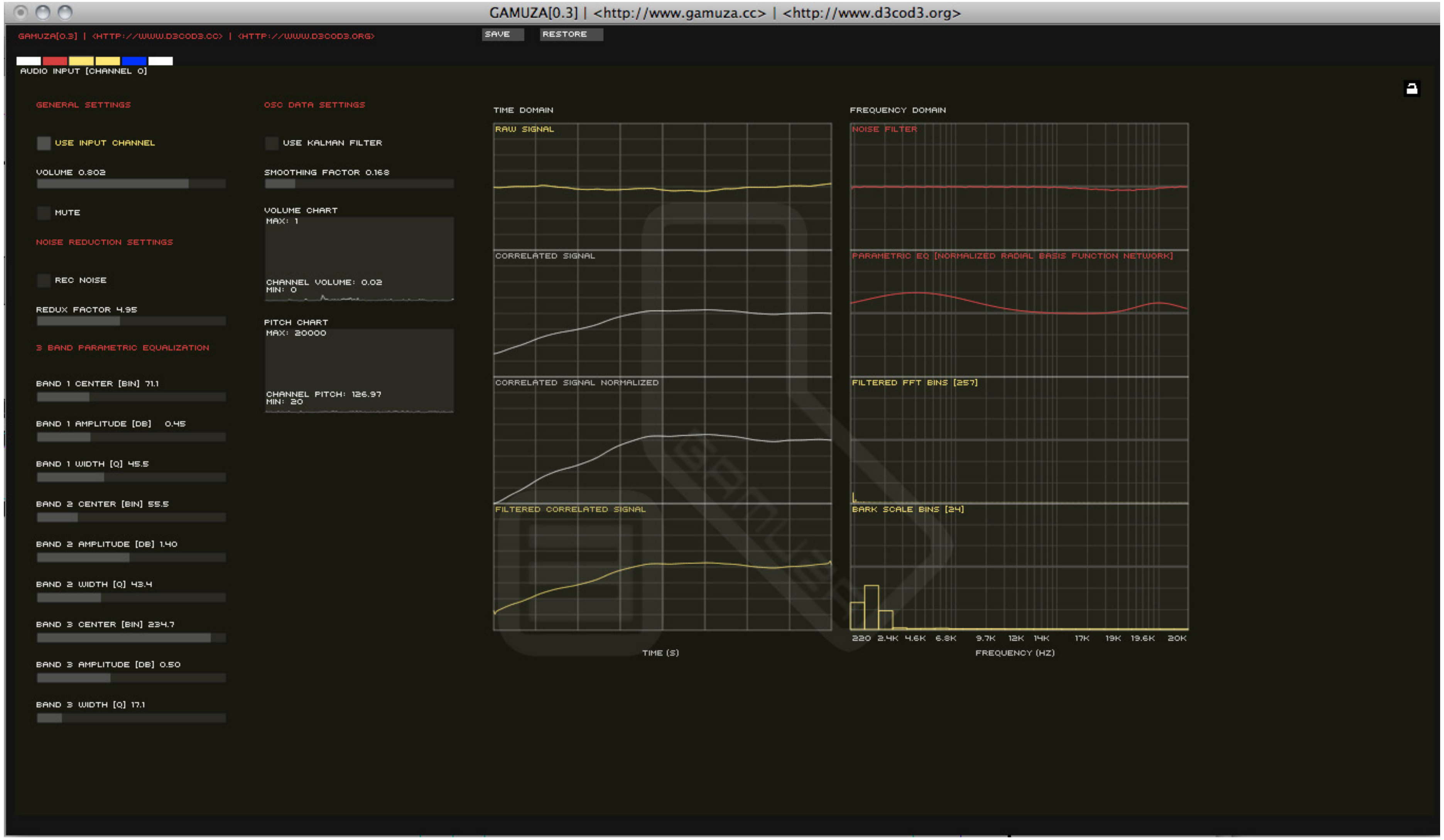
COMPUTE TRIGGER AREAS



```
TRIGGER AREAS: 0[0] 1[0] 2[0] 3[0] 4[0] 5[1] 6[0] 7[1] 8[0]
```

Video tracking module soon

Audio Input Module(s)


GAMUZA[0.3] | <http://www.gamuza.cc> | <http://www.d3cod3.org>

GAMUZA[0.3] | <HTTP://WWW.D3COD3.CC> | <HTTP://WWW.D3COD3.ORG>

SAVE RESTORE

AUDIO INPUT [CHANNEL 0]

GENERAL SETTINGS

USE INPUT CHANNEL

VOLUME 0.802

MUTE

NOISE REDUCTION SETTINGS

REC NOISE

REDUX FACTOR 4.95

3 BAND PARAMETRIC EQUALIZATION

BAND 1 CENTER [BIN] 71.1

BAND 1 AMPLITUDE [DB] 0.45

BAND 1 WIDTH [Q] 45.5

BAND 2 CENTER [BIN] 55.5

BAND 2 AMPLITUDE [DB] 1.40

BAND 2 WIDTH [Q] 43.4

BAND 3 CENTER [BIN] 234.7

BAND 3 AMPLITUDE [DB] 0.50

BAND 3 WIDTH [Q] 17.1

OSC DATA SETTINGS

USE KALMAN FILTER

SMOOTHING FACTOR 0.168

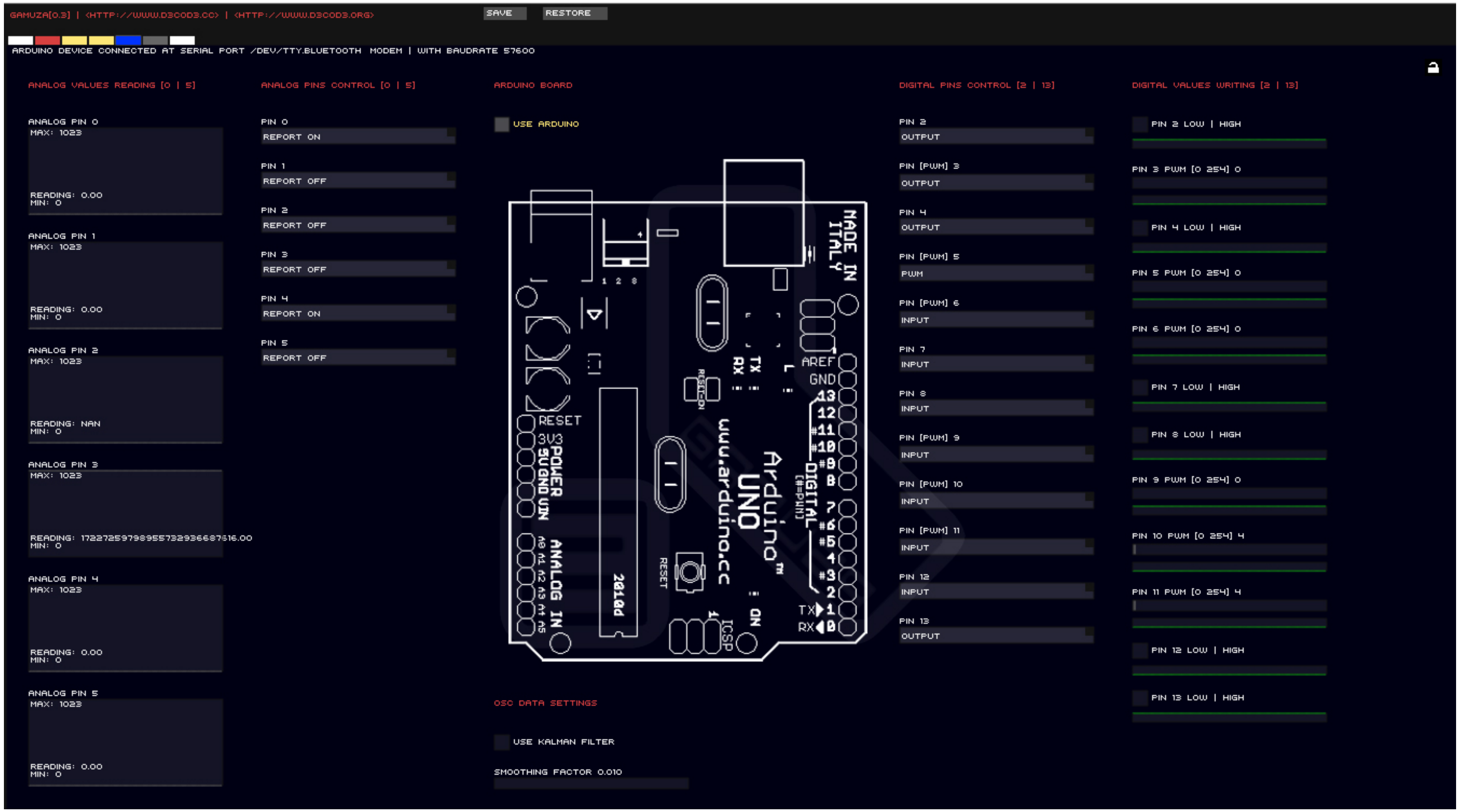
VOLUME CHART
MAX: 1
CHANNEL VOLUME: 0.02
MIN: 0

PITCH CHART
MAX: 20000
CHANNEL PITCH: 126.97
MIN: 20

TIME DOMAIN
RAW SIGNAL
CORRELATED SIGNAL
CORRELATED SIGNAL NORMALIZED
FILTERED CORRELATED SIGNAL
TIME (S)
FREQUENCY DOMAIN
NOISE FILTER
PARAMETRIC EQ [NORMALIZED RADIAL BASIS FUNCTION NETWORK]
FILTERED FFT BINS [257]
BARK SCALE BINS [24]
220 2.4K 4.6K 6.8K 9.7K 12K 14K 17K 19K 19.6K 20K
FREQUENCY (HZ)

Audio module manual, soon

Arduino Module


GAMUZA[0.3] | <HTTP://WWW.D3COD3.CC> | <HTTP://WWW.D3COD3.ORG>
ARDUINO DEVICE CONNECTED AT SERIAL PORT /DEV/TTY.BLUETOOTH MODEM | WITH BAUDRATE 57600

ANALOG VALUES READING [0 | 5]

ANALOG PIN 0
MAX: 1023
READING: 0.00
MIN: 0

ANALOG PIN 1
MAX: 1023
READING: 0.00
MIN: 0

ANALOG PIN 2
MAX: 1023
READING: NAN
MIN: 0

ANALOG PIN 3
MAX: 1023
READING: 17227259798955732936687616.00
MIN: 0

ANALOG PIN 4
MAX: 1023
READING: 0.00
MIN: 0

ANALOG PIN 5
MAX: 1023
READING: 0.00
MIN: 0

ANALOG PINS CONTROL [0 | 5]

PIN 0
REPORT ON

PIN 1
REPORT OFF

PIN 2
REPORT OFF

PIN 3
REPORT OFF

PIN 4
REPORT ON

PIN 5
REPORT OFF

ARDUINO BOARD

USE ARDUINO

OSC DATA SETTINGS

USE KALMAN FILTER

SMOOTHING FACTOR 0.010

DIGITAL PINS CONTROL [2 | 13]

PIN 2
OUTPUT

PIN [PWM] 3
OUTPUT

PIN 4
OUTPUT

PIN [PWM] 5
PWM

PIN [PWM] 6
INPUT

PIN 7
INPUT

PIN 8
INPUT

PIN [PWM] 9
INPUT

PIN [PWM] 10
INPUT

PIN [PWM] 11
INPUT

PIN 12
INPUT

PIN 13
OUTPUT

DIGITAL VALUES WRITING [2 | 13]

PIN 2 LOW | HIGH

PIN 3 PWM [0 254] 0

PIN 4 LOW | HIGH

PIN 5 PWM [0 254] 0

PIN 6 PWM [0 254] 0

PIN 7 LOW | HIGH

PIN 8 LOW | HIGH

PIN 9 PWM [0 254] 0

PIN 10 PWM [0 254] 4

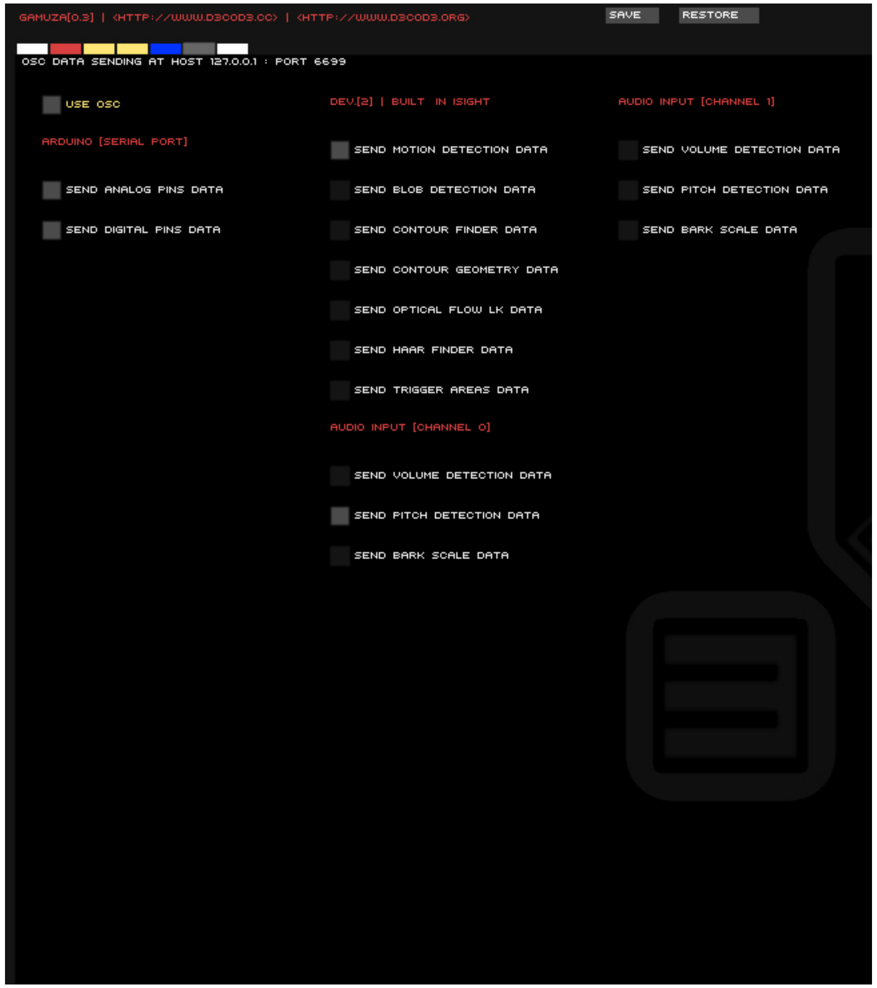
PIN 11 PWM [0 254] 4

PIN 12 LOW | HIGH

PIN 13 LOW | HIGH

Arduino module manual, soon

OSC Module



GAMUZA[0.3] | <HTTP://WWW.D3COD3.CC> | <HTTP://WWW.D3COD3.ORG>

SAVE RESTORE

OSC DATA SENDING AT HOST 127.0.0.1 : PORT 6699

USE OSC

ARDUINO [SERIAL PORT]

SEND ANALOG PINS DATA

SEND DIGITAL PINS DATA

DEV.[2] | BUILT IN ISIGHT

SEND MOTION DETECTION DATA

SEND BLOB DETECTION DATA

SEND CONTOUR FINDER DATA

SEND CONTOUR GEOMETRY DATA

SEND OPTICAL FLOW LK DATA

SEND HAAR FINDER DATA

SEND TRIGGER AREAS DATA

AUDIO INPUT [CHANNEL 0]

SEND VOLUME DETECTION DATA

SEND PITCH DETECTION DATA

SEND BARK SCALE DATA

AUDIO INPUT [CHANNEL 1]

SEND VOLUME DETECTION DATA

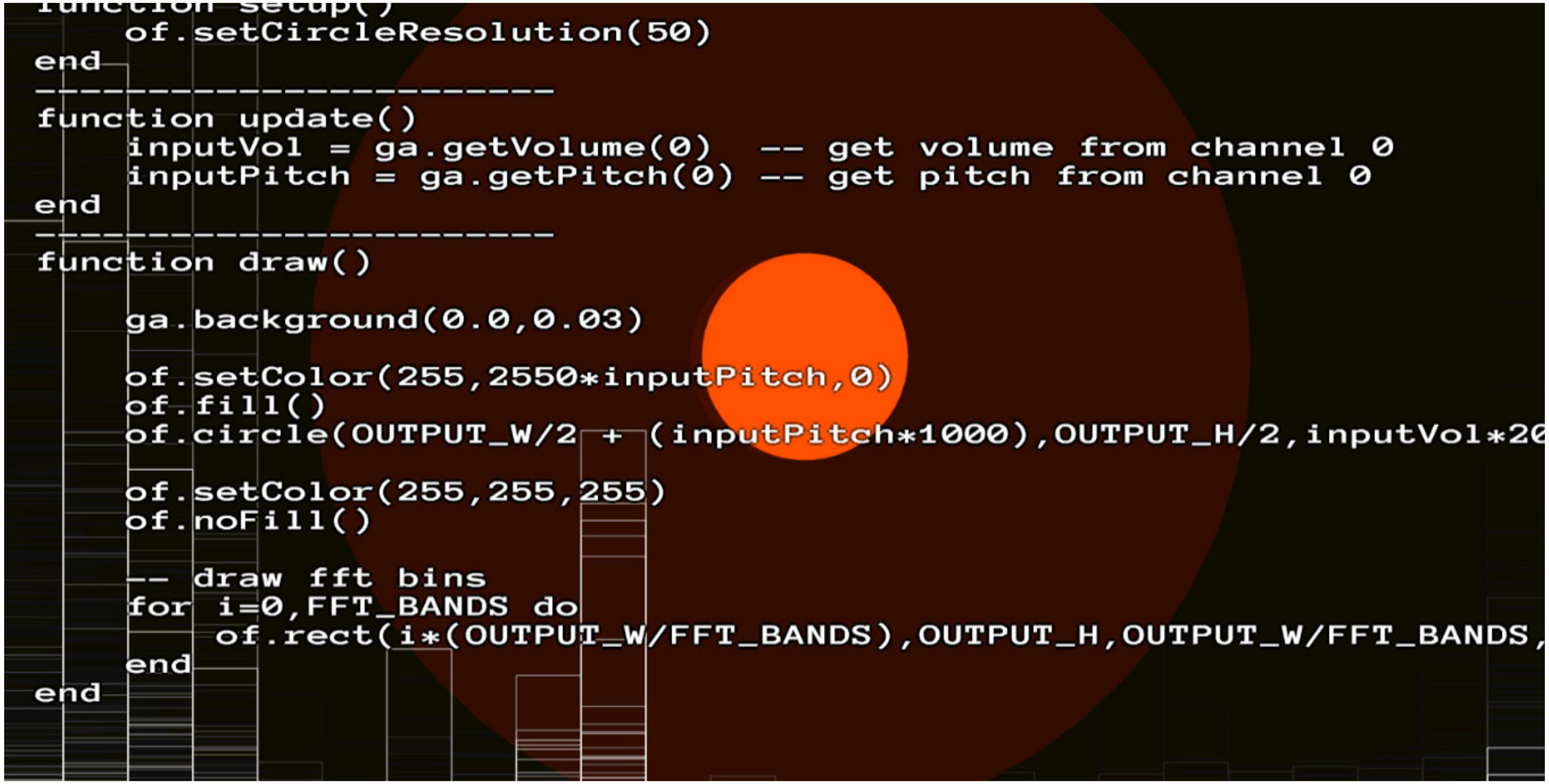
SEND PITCH DETECTION DATA

SEND BARK SCALE DATA

OSC module manual, soon

Scripting Language Reference

Look at the reference on project website:
http://gamuza.d3cod3.org/references/



```
function setup()
     of.setCircleResolution(50)
end
-----------------------------
function update()
     inputVol = ga.getVolume(0)   -- get volume from channel 0
     inputPitch = ga.getPitch(0)  -- get pitch from channel 0
end
-----------------------------
function draw()

     ga.background(0.0,0.03)

     of.setColor(255,2550*inputPitch,0)
     of.fill()
     of.circle(OUTPUT_W/2 + (inputPitch*1000),OUTPUT_H/2,inputVol*20

     of.setColor(255,255,255)
     of.noFill()

     -- draw fft bins
     for i=0,FFT_BANDS do
          of.rect(i*(OUTPUT_W/FFT_BANDS),OUTPUT_H,OUTPUT_W/FFT_BANDS,
     end
end
```

Keyboard Shortcuts

alt	f	:	toggle fullscreen
alt	j	:	toggle live coding mode
alt	r	:	render script
alt	d	:	open script [with file dialog]
alt	s	:	save script [directly to scripts folder]
alt	w	:	show/hide script
alt	t	:	show/hide timeline
alt	g	:	play/stop timeline
alt	x	:	cut to clipboard [OSX only]
alt	c	:	copy to clipboard [OSX only]
alt	v	:	paste from clipboard [OSX only]
ctrl	x	:	cut [internal editor]
ctrl	c	:	copy [internal editor]
ctrl	v	:	paste [internal editor]
alt	a	:	select all text
alt	e	:	clear editor
alt	b	:	blow up cursor
alt	o	:	save image frame
alt	p	:	print frame

LICENSE

❮❮❮ ❯❯❯

CREDITS

GAmuza v.03 'Raven Shammy'
----
http://gamuza.d3cod3.org

GAmuza is a HYBRID LIVE CODING Modular Application for interactive design development, performance & teaching

developed with OpenFrameworks 007[with some tweaks]
+
various code[sometimes modified] from different people:

[coders list]

Memo	Akten	<http://www.memo.tv/>
Zach	Gage	<http://www.stfj.net/>
Akira	Hayasaka	<http://www.ampontang.com/en>
Golan	Levin	<http://www.flong.com>
Zachary	Lieberman	<http://www.thesystemis.com/>
Rui	Madeira	<http://www.rui-m.com/>
Kyle	McDonald	<http://kylemcdonald.net/>
Damian	Stewart	<http://www.frey.co.nz/>
Chris	Sugrue	<http://www.csugrue.com>
Theo	Watson	<http://muonics.net/>
Dan	Wilcox	<http://danomatika.com>

Idea&Development by Emanuele Mazza.

GAMUZA