

GAmuza. Hybrid live coding/modular application

Contenidos

1. Qué es GAmuza	5
1.1. Premisas	7
1.2. Instalación	8
1.3. ScriptEditor	11
1.4. Comandos de teclado	12
2. Interface. Módulos de aplicación	13
2.1. Setting Manager	13
2.2. Live Coding	22
2.2.1. Timeline	26
2.3. Análisis Video	27
2.3.1. Computer vision	30
2.3.2. Sensor Kinect	36
2.4. Análisis audio	41
2.4.1. Audio Stream (Input/Output)	44
2.5. Interfaces físicas	49
2.5.1. Arduino	52
2.6. Comunicación OSC	53

GAmuza. Hybrid live coding/modular application. rel 04

© 2012 Emanuele Mazza

Texto: Emanuele Mazza & María José Martínez de Pisón

Dibujos: Carlos Maiques

Valencia, España.

Este manual forma parte de un libro en preparación sobre el entorno de desarrollo GAmuza

Algunos derechos reservados. Reconocimiento - Compartir Igual (by-sa): Se permite el uso



comercial de la obra y de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

1. Qué es GAmuza

GAmuza es un software híbrido que conjuga un entorno ScriptEditor con distintas aplicaciones modulares para el desarrollo de diseño interactivo, la realización de performances audiovisuales en directo y principalmente la enseñanza de arte interactivo o electrónico. Es *open-source*, se distribuye bajo licencia MIT¹ y está desarrollado para Linux Ubuntu x86 y para Mac OSX 10.6 o superior. Puede descargarse en: <http://www.gamuza.cc>

Live-Coding

El núcleo central de GAmuza es el entorno Live-coding para hacer prototipos gráficos, performances o enseñar programación visualizando las modificaciones del código de forma casi inmediata. Está basado en el lenguaje de scripting integrable Lua², (utilizando la librería Luabind³ y una versión modificada de ofxLua⁴) y comprende toda la API OpenFrameworks⁵ v.0071, estableciendo conexiones de inputs/outputs entre las funciones y los módulos de aplicación.

Sistema modular

GAmuza comprende distintas aplicaciones modulares fácilmente configurables a través de GUI. Cada uno de estos módulos puede activarse, o no, en función del tipo de trabajo a realizar y para reducir el consumo del equipo. Los módulos activos enlazan directamente con las funciones y variables de salida del **ScriptEditor**. Actualmente los módulos son:

- Computer vision
- Sensor Kinect
- Audio Stream (input/output)
- Arduino
- OSC

¹ MIT License <<http://mit-license.org/>> [11.05.2012]

² The programming language Lua <<http://www.lua.org/>> [11.05.2012]

³ Luabind es una librería que ayuda a crear enlaces entre C++ y Lua. Ver <http://www.rasterbar.com/products/luabind.html> [12.05.2012]

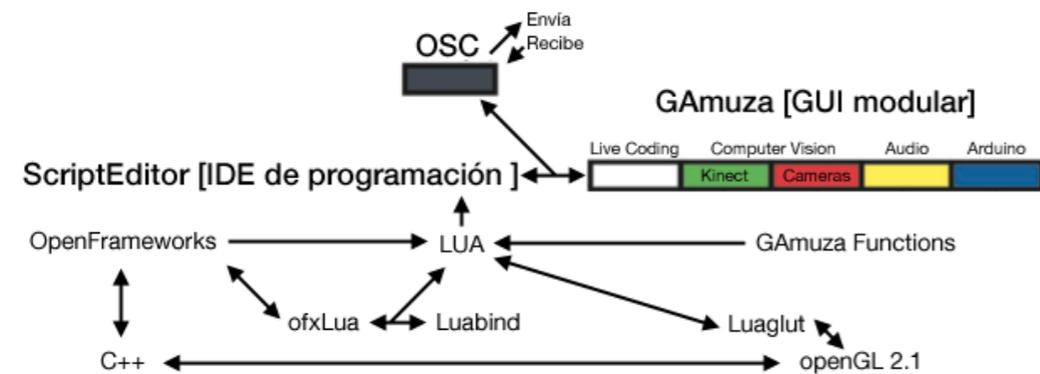
⁴ ofxLua es un addon de Open Frameworks para que funcionen scripts Lua incrustados dentro de una aplicación OpenFrameworks. Ver <https://github.com/danomatika/ofxLua> [12.05.2012]

⁵ openFrameworks <<http://www.openframeworks.cc/>> [12.05.2012]

En resumen, GAmuza es un software que recoge y coordina de forma particular lenguajes (Lua y C++) y plataformas (openFrameworks y openGL) ya existentes, e incorpora otras librerías propias para facilitar los primeros pasos de estudiantes y artistas en el campo de la programación creativa. Por otra parte, combina las dos vías habituales de programación: consola de código textual (ScriptEditor) vinculada a las aplicaciones modulares que se ajustan mediante GUI, para que ese proceso de aprendizaje creativo sea más fácil, pero sin ocultar las estructuras de programación. Para hacer funcionar esta hibridación se han programado librerías específicas de GAmuza, que como veremos más adelante, se diferencian sintácticamente de las retomadas de otras plataformas. Retomando un término de Armando Montesinos, GAmuza más que un collage de lenguajes y plataformas, reconstruye un entorno de programación por medio de “zurcidos”, utiliza parches y genera tejido nuevo entrecruzando funciones, sin ocultar los fragmentos.

La facilidad de uso de GAmuza reside en la simplificación del lenguaje Lua (en sí bastante ligero) y la comunicación de dispositivos externos por los módulos GUI, manteniendo la estabilidad y potencia del lenguaje C++.

El lenguaje de programación de GAmuza aglutina el uso de código en lenguaje Lua⁶ con funciones propias de openFrameworks⁷ y otras funciones específicas de GAmuza que comunican directamente con los datos de entrada y salida de las interfaces gráficas de los módulos, es importante entender esta movilidad o flexibilidad de niveles y lenguajes.



6 The language programming Lua <<http://www.lua.org/>> [11.06.2012]

7 openFrameworks <<http://www.openframeworks.cc/documentation/>> [10.06.2012]

1.1. Premisas

Dilema GUI versus Code

En el campo de la enseñanza del arte digital es común ver posiciones que defienden el uso de software libre, pero entre ellos se plantea otro dilema: el uso de software de programación gráfica, como Pure data, o enseñar código, línea de comandos, como Processing.

Hay un complicado entramado de relaciones y significancias entre ambas posturas que a veces parecen responder a campañas de índole casi publicitaria: gráficos-amigables versus entornos de programación-hostil. El adjetivo tras el guión indica el tono y posicionamiento del discurso. De nuevo el debate entre imagen y texto (textual turn⁸ vs. pictorial turn⁹), y como hemos señalado antes, GAmuza opta por la hibridación.

Las interfaces gráficas de usuario (GUI) de los módulos están programadas y diseñadas para facilitar procesos de comunicación con dispositivos de audio, vídeo o Arduino, pero no para ocultar conceptos o estructuras de programación, por eso incorpora el ScriptEditor, estableciendo una situación híbrida que aúna GUI y Code.

Facilitar, para que los estudiantes, artistas o gente interesada no experta pierda el miedo a algo que comúnmente se califica como demasiado complicado. Pero lo complicado o difícil es solo lo que no se comprende. Por eso....

Tecnología para la gente

...GAmuza no oculta los conceptos o estructuras. Comprender las herramientas que usamos todos los días, es el primer paso para no ser esclavo de la tecnología.

Más allá del tipo de relación que los estudiantes de arte mantienen con las tecnologías de información, la gente en general está multiplicando el tiempo y tipo de usos con estos medios.

Se ha hecho popular publicar ideas personales, intereses, conversaciones, gustos, opiniones, fotos, vídeos... Todo el mundo está conectado, desde su portátil o smartphone, y en 5 minutos te hacen sentir que eres el webmaster de todas tus ficciones o vidas virtuales diferentes, constantemente online con todos tus amigos virtuales, haciendo todo esto sin conocer la tecnología que estás utilizando, y esto puede significar muchas cosas, una de ellas es “gente para la tecnología”, o en otras palabras, los medios están preparando “gente a la que vender tecnología”. GAmuza es sólo un software, pero creado con un concepto en mente: “tecnología para la gente”, no lo contrario.¹⁰

8 Rosalind Krauss (1966), “Welkome to the Cultural Revolution” en *October*, Vol. 77 [artículo on-line] <<http://www9.georgetown.edu/faculty/irvinem/theory/Krauss-October-77-1996-WelcomeToTheCulturalRevolution.pdf>> [30.08.2012]

9 Liderado en gran medida por W. J. T. Mitchell <<http://humanities.uchicago.edu/faculty/mitchell/home.htm>> [30.08.2012]

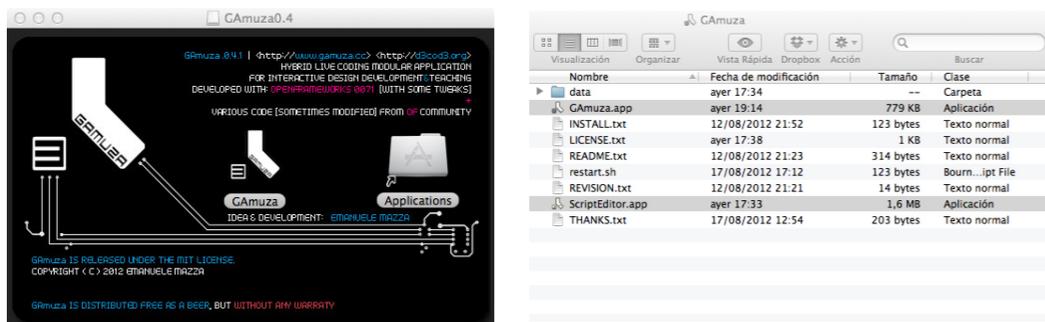
10 Mazza, Emanuele, “Why GAmuza” [texto on-line] <<http://gamuza.d3cod3.org/about/concept/>> [11.05.2012]

1.2. Instalación

GAmuza está programado para funcionar en mac OSX desde 10.6 (Snow Leopard) y en Ubuntu Linux x86, 32 bits.

Mac OSX

Desde <<http://gamuza.d3cod3.org>> descargar la aplicación, abrir (montar) el archivo .dmg y arrastrar el icono de GAmuza al acceso directo de la carpeta Aplicaciones, tal como indica la flecha en la imagen. El proceso de instalación ha terminado.



Si abrimos la carpeta de GAmuza veremos que contiene dos aplicaciones: ScripEditor.app y GAmuza.app

Ubuntu Linux x86

Descargar el archivo .tar.gz desde <<http://gamuza.d3cod3.org>>, abrir la ventana del Terminal y escribir:

```
tar xvfz GAmuza-xxxx-Linux.tar.gz
-- sustituid -xxxx- por el n° de la versión descargada de GAmuza
```

Esto creará una carpeta llamada GAmuza, después cambiamos al directorio

```
cd GAmuza
```

Abrir la carpeta install y ejecutar el archivo

```
cd install && sudo sh 00_install_dependencies.sh
```

Después:

```
sudo sh 01_get_GAmuza_source.sh
```

En algunas versiones de Linux a veces aparece un error relacionado con el archivo /usr/include/c++/4.5/complex;

En tal caso, editar el archivo:

```
sudo gedit /usr/include/c++/4.5/complex
```

Ir a la línea 953, donde está la función template, inline complex, pow

```
#ifndef __GXX_EXPERIMENTAL_CXX0X__
// _GLIBCXX_RESOLVE_LIB_DEFECTS
// DR 844. complex pow return type is ambiguous. template
inline complex
pow(const complex& __z, int __n)
{ return std::__pow_helper(__z, __n); }
#endif
```

y comentar la función completa

```
#ifndef __GXX_EXPERIMENTAL_CXX0X__
// _GLIBCXX_RESOLVE_LIB_DEFECTS
// DR 844. complex pow return type is ambiguous. /* template
inline complex
pow(const complex& __z, int __n)
{ return std::__pow_helper(__z, __n); }*/
#endif
```

Después, ejecutar 02_install_GAmuza.sh

```
- sudo sh 02_install_GAmuza.sh
```

GAmuza tiene una configuración de aplicaciones diferente en Mac OSX y Linux x86. Las principales diferencias son:

ScriptEditor:

- En Mac hay una aplicación independiente para escribir el código.
- En Linux está integrado en el módulo Live Coding. Puede escribirse el código directamente ahí, o usar cualquier editor de texto, guardar el script como archivo .txt, .lua, o .ga y abrirlo desde el botón "Load script" situado en la parte inferior de Live Coding, igualmente pueden guardarse los scripts desde "Save script".

Setting Manager.

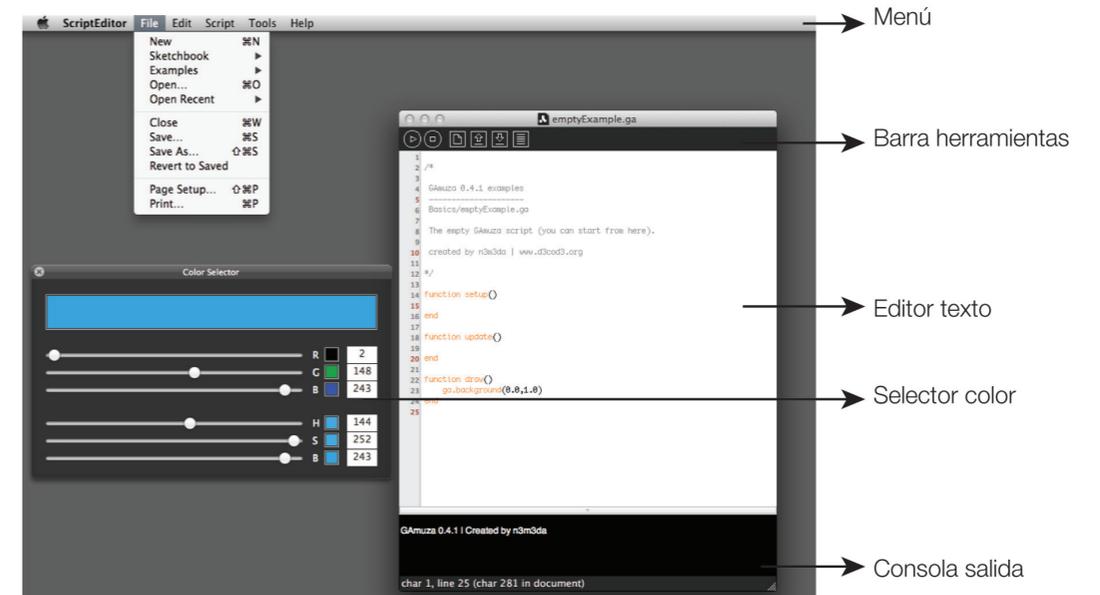
- En Linux es una aplicación independiente que se ajusta antes de iniciar GAmuza, si previamente no se había configurado el proyecto.
- En mac está incluida como una pestaña más de los módulos. Tras configurar el proyecto debe reiniciarse GAmuza

Teclas de comando.

- Hay atajos de teclado para Mac y Linux, y otros específicamente para Linux

1.3. ScriptEditor

ScriptEditor solo está disponible en Mac. Al abrir la aplicación se activan tanto el editor de texto como GAmuza. Las dos son independientes pero vinculadas por OSC. Puede cerrarse una de ellas y la otra seguirá activa.



La interface del ScriptEditor sigue directamente el modelo marcado por el PDE Processing, como epígono de una de sus influencias directas. Principalmente aporta un editor de texto para el código que en su parte superior dispone un menú y barra de herramientas, y en su parte inferior una consola de salida.

En el menú desplegable superior se encuentra: File (abrir, guardar, cerrar, imprimir...), Edit (deshacer, rehacer, cortar, copiar, pegar, borrar, seleccionar todo, encontrar, reemplazar...), Script (enviar a GAmuza, borrar consola), Tools (selector de color). Y en la barra de herramientas unos botones para facilitar las labores usuales: enviar a GAmuza, limpiar script, nuevo script, abrir script y guardar como y limpiar consola.

El editor de texto tiene una columna numerada a la izquierda para facilitar la localización de errores. También asigna una serie de colores a términos que reconoce vinculados a:

- Funciones y términos de GAmuza: naranja `of.setColor(255)`
- Constantes, azul `OUTPUT_W`
- OpenGL: funciones, azul turquesa y constantes: marrón pardo `glBegin(GL_QUADS)`
- Strings (cadena caracteres entrecomillados): rojo `"texto entrecomillado"`
- Comentarios: gris `// comentario`
- Negro para todo lo demás. `x = 125`

1.4. Comandos de teclado

Para compatibilizar y optimizar las posibilidades de GAmuza se han programado comandos de teclado, es recomendable conocerlos antes de utilizar las interfaces de los módulos de aplicación. La combinación se realiza prioritariamente con la tecla Alt para evitar duplicidades con los sistemas operativos.

Mac OSX y Linux

- Alt+f.** Poner/quitar la aplicación a pantalla completa. Para que se active la segunda pantalla es necesario que esté en pantalla completa.
- Alt+j.** Amplia/reduce el tamaño de ventana del previo en la interface Live coding.
- Alt+w.** Mostrar/ocultar el script sobre la ventana de salida o el previo ampliado.
- Alt+t.** Mostrar/ocultar el Timeline
- Alt+g.** Reproducir/Parar el Timeline
- Alt+o.** Guardar el frame actual. Lo guarda en la carpeta GAmuza/data/export/frames y lo nombra como "frame0_día_mes_año.tif", y si repetimos las teclas: "frame1...", "frame2..."
- Alt+p.** Imprimir el frame actual, hace falta una impresora conectada al ordenador.

Sólo en Linux

- Alt+r.** Procesa el script. Hay también un botón en la interface para hacerlo.
- Alt+d.** Abrir un script guardado (aparece un cuadro de diálogo para seleccionarlo)
- Alt+s.** Guardar el script, lo hace directamente en la carpeta GAmuza/data/scripts con el nombre "gamuzaScript_XXXXXXXXX.lua" donde las x corresponden a: año mes día hora minutos segundos
- Ctrl+x.** Cortar (información solo utilizable en la propia consola live coding)
- Ctrl+c.** Copiar (información solo utilizable en la propia consola live coding)
- Ctrl+v.** Pegar (información que proviene de la propia consola live coding)
- Alt+a.** Seleccionar todo el texto
- Alt+e.** Limpiar editor (borrar todo el código)
- Alt+b.** Ampliar el tamaño del cursor momentáneamente para verlo con facilidad.

2. Interface. Módulos de aplicación

En realidad, la tarea de descifrar las historias una por una me ha hecho descuidar hasta ahora la peculiaridad más saliente de nuestro modo de narrar, a saber, que todo relato corre al encuentro de otro relato y mientras un comensal avanza en su fila, desde la otra punta avanza otro en sentido opuesto, porque las historias contadas de izquierda a derecha o de abajo hacia arriba pueden también leerse de derecha a izquierda o de arriba hacia abajo, y viceversa.¹¹

Italo Calvino

Cuando se abre GAmuza, desde ScriptEditor o directamente, la primera interface que aparece es el módulo Live coding. En su parte superior hay una serie de rectángulos de color, estos rectángulos son las pestañas que indican los módulos que están activos y dan acceso a ellos. Tal como advierte Italo Calvino, toda estructura modular puede describirse siguiendo direcciones distintas.



2.1. Setting Manager

Vamos a empezar a describir la última pestaña blanca de la versión para Mac OSX que corresponde al Setting Manager (en linux la aplicación SettingManager.app es independiente y está en la carpeta de GAmuza). Con esta interface se activan o desactivan los módulos y se ajusta su configuración.

No es necesario, ni recomendable, tener módulos activos si no se van a utilizar, porque supone un consumo innecesario de los recursos del sistema. Al abrir GAmuza, un buen hábito de inicio es ir al Setting Manager, ajustar la configuración de GAmuza a las características del ordenador y las necesidades del proyecto.

Pasamos a describir la interface del Setting Manager, a partir de su imagen general y analizando después paso a paso los elementos configurables de cada módulo.

¹¹ Italo Calvino (1993), *El Castillo de los destinos cruzados*. Madrid: Siruela, pág. 37.

GAMUZA [0.4.1] 'SILENT SHORE' BETA | <HTTP://WWW.GAMUZA.CC> | <HTTP://WWW.D3COD3.ORG> SAVE LOAD

GAMUZA SETTING MANAGER

RESTART

MODULES

SENSOR KINECT

COMPUTER VISION

AUDIO STREAM (INPUT/OUTPUT)

ARDUINO

SCREENS

MAIN SCREEN
1680x1050 (16:10)

SECONDARY SCREEN
1024x768 (4:3)

OUTPUT VIDEO MAPPING

MAPPING GRID RESOLUTION 8

AUTOMATION (INSTALLATION MODE)

AUTO FULLSCREEN

AUTOLOAD SCRIPT ON START

CHOOSE SCRIPT

COMPUTER VISION

USE TESTING VIDEO

CHOOSE VIDEO

CHOOSE HAAR FINDER FILE

SENSOR KINECT

USE INFRARED

LED STATE
off

AUDIO STREAM (INPUT/OUTPUT)

AUDIO DEVICE
Apple Inc.: Built-in Microphone

SAMPLING RATE
44100

BUFFER SIZE
512

MIDI INPUT

MIDI DEVICE
Vestax PC-CONTROLLER

ARDUINO

BAUDRATE
57600

SERIAL PORT
/dev/tty.usbmodemFa141

OSC ENGINE

SENDING TO IP
192.168.1.2

SENDING TO PORT
9666

RECEIVING AT PORT
8997

events logger

```

-----
MAX_FBO_NUMLSAMPLES: 8
-----
SERIAL DEVICES
-----
/dev/tty.usbmodemFa141
/dev/cu.usbmodemFa141
/dev/tty.Bluetooth-Modem
/dev/cu.Bluetooth-Modem
/dev/tty.Bluetooth-PDA-Sync
/dev/cu.Bluetooth-PDA-Sync
-----
AUDIO DEVICES:
-----
device: 0 (Apple Inc.: Built-in Microphone)
output channels: 0
input channels: 2
sample rates: 8000,9600,11025,16000,22050,32000,44100,48000,88200,96000,
-----
device: 1 (Apple Inc.: Built-in Input)
output channels: 0
input channels: 2
sample rates: 8000,9600,11025,16000,22050,32000,44100,48000,88200,96000,
-----
device: 2 (Apple Inc.: Built-in Output)
output channels: 2
input channels: 0
sample rates: 44100,48000,88200,96000,
-----
MIDI DEVICES:
-----
device: 0 (Vestax PC-CONTROLLER)
-----

```

El primer grupo de opciones se refiere a los módulos GUI, simplemente hay que seleccionar o deseleccionar los módulos que se quieren activar o desactivar, clicando en el cuadrado que hay a la izquierda de los nombres. Las opciones son.

MODULES

```

SENSOR KINECT           // tracking con el sensor Kinect

COMPUTER VISION         // con otras cámaras que reconozca el ordenador

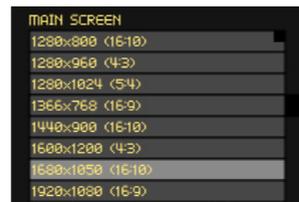
AUDIO STREAM (INPUT/OUTPUT) // análisis de entrada y/o salida de audio

ARDUINO                 // trabajar con el microcontrolador Arduino
  
```

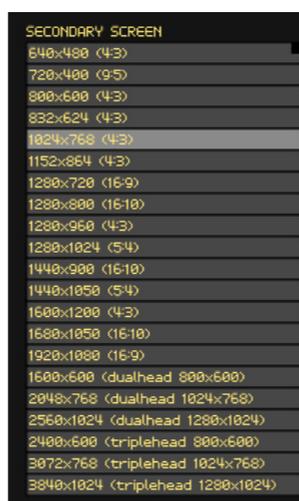
IMPORTANTE, si se activa el módulo Sensor Kinect, por ejemplo, y no está la Kinect conectada al ordenador, muy probablemente GAmuza se cerrará al no encontrar el dispositivo. Lo mismo se recomienda para los otros módulos, no deben activarse si no están los dispositivos apropiados conectados.

SCREENS

MAINSCREEN: abre un menú desplegable para seleccionar la resolución de la pantalla principal. La resolución mínima para visualizar GAmuza es 1280x 800 píxeles.



SECONDARY SCREEN: menú desplegable para seleccionar la resolución de la segunda pantalla, (se corresponde al tamaño de la ventana de salida). En ambos casos, elegir según la resolución de los monitores o proyectores que se utilicen. Las últimas opciones hacen referencia al uso de tarjetas que duplican o triplican el tamaño de la ventana de salida. Si hay una de estas tarjetas conectada y el ordenador la reconoce, GAmuza genera una ventana de salida para sus dimensiones.



Si se cambian las dimensiones de la segunda pantalla hay que resetear los ajustes del mapping en el módulo de Live Coding. O cargar un archivo de mapping si previamente se había guardado para las nuevas dimensiones de pantalla.

OUTPUT MAPPING

GAmuza genera una rejilla sobre la imagen de salida para proyecciones de mapping. Con un slider se puede elegir el número de nodos de esa rejilla, desde un mínimo de 1 (significa que el rectángulo de la proyección tiene un punto de ajuste activo en cada vértice, con los que se puede deformar la proyección), a un máximo de 20, (la imagen tiene 400 rectángulos ajustables).

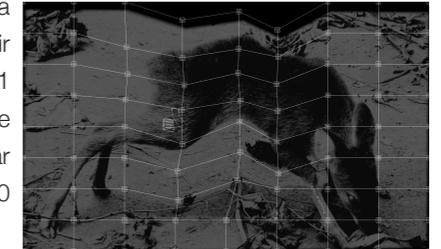
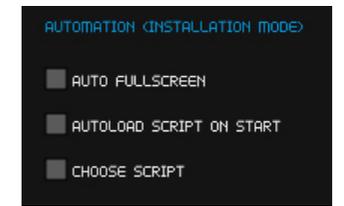


Imagen con 8 nodos de mapping, 64 rectángulos ajustables

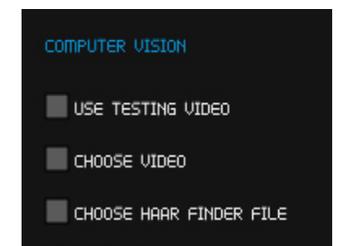
AUTOMATION (INSTALLATION MODE)

Opción pensada para la presentación de instalaciones. Permite seleccionar un script, "Choose script", abriendo una ventana de diálogo para elegirlo dentro del ordenador. Si se activa "Autoload script on start" GAmuza se inicia con ese script. Y por último, si se activa "Auto fullscreen" al abrirse el programa, está directamente a pantalla completa y procesando el script seleccionado.



COMPUTER VISION

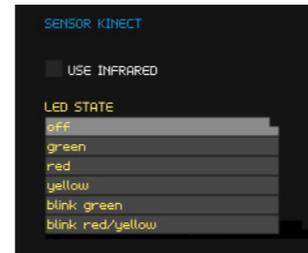
Estas opciones se ajustan si el módulo Computer Vision está seleccionado. **USE TESTING VIDEO** ofrece la posibilidad de utilizar un archivo de video como test, es decir, **desactiva la webcam** o cámaras de video, y utiliza los algoritmos y ajustes de tracking video sobre el archivo de video seleccionado con **CHOOSE VIDEO**.



La opción **CHOOSE HAAR FINDER FILE**, permite elegir el algoritmo de reconocimiento de partes del cuerpo. Por defecto, GAmuza carga el reconocimiento de rostro frontal. Para cambiarlo, al clicar esta opción se abre una ventana de diálogo, y se puede seleccionar en la carpeta: **GAMUZA/DATA/SETTINGS/HAARXML**, o donde uno tenga ficheros .xml de haar cascade.

SENSOR KINECT

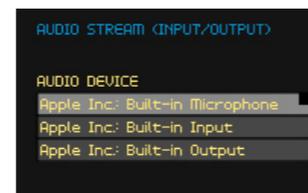
Esta sección solo se utiliza si se ha activado el módulo Sensor Kinect. Las opciones son simples, se puede activar el uso de visión infrarroja, si no kinect utilizará visión normal. En el menú desplegable de abajo se puede desactivar el led frontal del sensor kinect (OFF), y si quiere utilizarse, se puede elegir el color del Led y si parpadea o no:



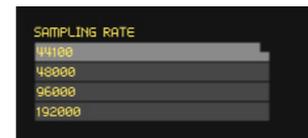
AUDIO STREAM (INPUT/OUTPUT)

Esta sección se ajusta solo si el módulo del mismo nombre está activado.

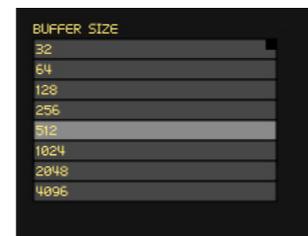
Lo primero es seleccionar **AUDIO DEVICE**, el dispositivo de audio con el que trabajará GAmuza. El menú desplegable muestra los que están accesibles en el ordenador.



En la opción **SAMPLING RATE** (Frecuencia de muestreo) hay que seleccionar una acorde con el dispositivo: en la consola de texto del Setting Manager aparecen todos los dispositivos de audio que reconoce el sistema con el Sample Rate que soportan.



En **BUFFER SIZE** seleccionar el tamaño de Buffer con que se va a trabajar (cantidad de muestras almacenadas), 512 es un tamaño válido para la mayoría de los proyectos, si se trabaja con los datos de FFT puede requerir un tamaño mayor, hay que tener en cuenta que a mayor buffer más recursos de procesador consume.

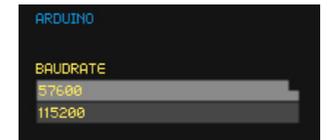


Algunos ordenadores tienen la entrada y salida de audio por separado, si se selecciona el dispositivo de entrada, no se pueden generar sonidos de salida. La mayoría de las tarjetas de audio presentan la opción de entrada y salida juntas, por lo que permiten hacer ambas.

ARDUINO

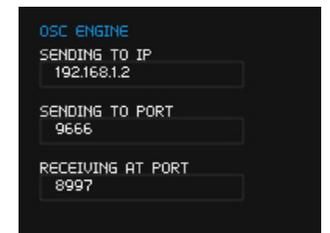
Como en los casos anteriores, esta sección del Setting Manager solo se ajusta si el módulo de Arduino está seleccionado.

BAUDRATE (57600 por defecto) y **SERIAL PORT** (el puerto serie). Puede utilizarse el software de Arduino para comprobar mediante el Serial Monitor cuál es el puerto asignado y seleccionar en GAmuza el mismo.



OSC ENGINE

Ajustes para el protocolo de transferencia de datos OSC, se introducen directamente en el campo de texto. Los dos primeros **SENDING TO IP** y **SENDING TO PORT** envían datos a un ordenador con el número de IP y puerto que se pongan. El último, **RECEIVING AT PORT** indica el puerto de este ordenador que recogerá los datos enviados desde otro.



En linux estas opciones se pueden modificar en el archivo de configuración settingManager.xml

MIDI

Aunque no existe un módulo de MIDI, se pueden recoger datos de estos dispositivos y hay funciones específicas de GAmuza para comunicar con ellos.



Tras seleccionar las opciones necesarias hay que guardar la configuración clicando Save en la parte superior de la interface. Diferencias entre Mac y Linux:

Mac: se abrirá un cuadro de diálogo para seleccionar la carpeta donde se quieran guardar los ajustes, GAmuza genera un archivo XML con la configuración. Antes de empezar a trabajar con GAmuza, clicar Restart (botón superior izquierdo) para que se reinicie el programa con los ajustes realizados. También puede cargarse un archivo de configuración que ha sido previamente guardado.

Linux: se guardan los cambios directamente en el archivo gamuzaSettings.xml. Cerrar la aplicación SettingManager.app y volver a abrir GAmuza.

GAMUZA [0.4.1] 'SILENT SHORE' BETA | <HTTP://WWW.GAMUZA.CC> | <HTTP://WWW.D3C003.ORG> SAVE LOAD

LIVE CODING [SECOND SCREEN FULLSCREEN OUTPUT @ 1440 x 900]

MAPPING SETTINGS

- DRAW MAPPING GRID
- RESET MAPPING POINTS
- LOAD MAPPING SETTING
- SAVE MAPPING SETTING

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 B&W 0.00

FILM WHITE EXPOSURE 0.0

FILM WHITE DIFFUSION 0.00

VIDEO OUTPUT SETTINGS

- ACTIVATE SECONDARY SCREEN
- USE VERTICAL SYNC

FPS
MAX: 200
FPS: 25.29
MIN: 1

2.2. Live Coding

Los algoritmos son pensamientos, las motosierras son herramientas¹².

Live coding significa programar en directo, un término vinculado tanto a los entornos de programación con un tiempo de compilación *infraleve*¹³ —‘just in time programming’ o ‘real time’—, como a la realización de directos de audio y/o imagen que utilizan esos entornos, y donde la presencia del código como imagen o sobre la imagen que generan, es uno de los rasgos que lo caracterizan.

Los software de Live Coding, Fluxus¹⁴ y Zajal¹⁵, fueron los que más influyeron en el desarrollo de GAmuza. Fluxus es un entorno de programación en tiempo real basado en C++ / Scheme y desarrollado por Artem Baguinski. Zajal es un lenguaje de programación creativo, basado en Ruby y que funciona bajo openframeworks, desarrollado por Ramsey Nasser.

Siguiendo sus pasos, en las primeras versiones de GAmuza, hasta rel.0398, el código se escribía directamente en el módulo de Live Coding, y aun es así en GAmuza para Linux. En la versión 041 para Mac OSX el sistema Live coding se hibrida un paso más, el código puede mostrarse en la ventana de salida, pero cuenta con un editor independiente, combinando las necesidades de las performances en directo con su uso para la enseñanza de programación.

```
*/
freq = 0.0
noiseTime = {}
step = {}
mapNoise = {}

function setup()
  for i=0,OUTPUT_W do
    noiseTime[i] = of.random(0,10000)
    step[i] = of.random(0.001,0.003)
    mapNoise[i] = 0
  end
end

function update()
  for i=0,OUTPUT_W do
    mapNoise[i] = of.map(of.noise(noiseTime[i]),0.15,0.85,0.0,1)
    noiseTime[i] = noiseTime[i] + step[i]
  end
end

function draw()
  ga.background(0.0,0.01)

  of.pushMatrix()
  of.translate(0, OUTPUT_H/2, 0)
  of.noFill()

  of.beginPath()
  freq = ga.getPitch(0)*20000
  phase = 0
  of.setColor(29 - freq/30,179 - freq/30,211 - freq/30,freq/30)
  for i=0,OUTPUT_W do
```

En el módulo de Live Coding existen también opciones de ajuste y, como en la interface anterior, pasamos a describirlas individualmente.

¹² Stephen Ramsay, Algorithms are Thoughts, Chainsaws are Tools. [video on-line] <<https://vimeo.com/9790850>> [04.09.2012]

¹³ Infraleve, término de Duchamp.

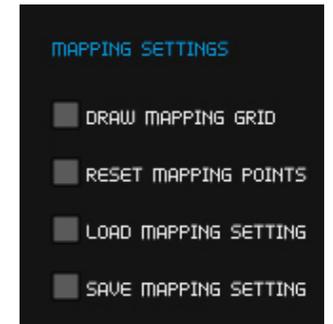
¹⁴ Fluxus) <<http://www.pawfal.org/fluxus/>> [03.09.2012]

¹⁵ Zajal, <<http://zajal.cc/>> [03.09.2012]

MAPPING SETTING

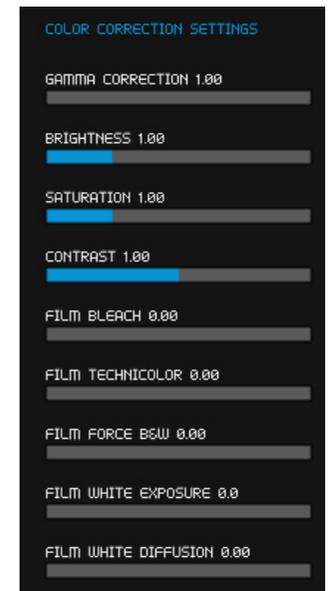
Hace referencia a la rejilla de mapping con las siguiente opciones:

- Activa/desactiva que se muestre la rejilla
- Resetea los cambios (vuelve al estado inicial)
- Abrir una configuración previamente guardada.
- Guardar la configuración del mapping



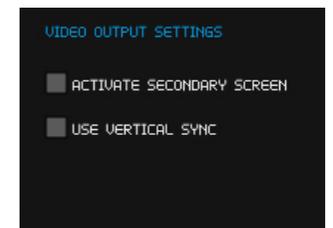
COLOR CORRECTION SETTINGS Ajustar ecolor y aplicar efectos:

- **GAMMA CORRECTION:** ajusta la “sensación” de contraste de la imagen si se percibe blanquecina o “lavada”.
- **BRIGHTNESS:** ajusta el brillo de la imagen.
- **SATURATION:** satura o desatura el color de la imagen.
- **CONTRAST:** ajusta el contraste.
- **FILM BLEACH:** emulación de una técnica de revelado analógico que produce contraste alto y saturación baja. También llamado Bleach Bypass.
- **FILM TECHNICOLOR:** emulación del rango de color de los TV de tubo catódico de los 80’s.
- **FILM FORCE B&W:** Los tres últimos filtros están relacionados y el ajuste de uno repercute en los otros. Este aporta otra forma para desaturar.
- **FILM WHITE EXPOSURE:** ajusta la sensación de luz.
- **FILM WHITE DIFFUSION:** emulación del uso de filtros de gelatina White Diffusion para suavizar sombras.



VIDEO OUTPUT SETTINGS

- Activa/desactiva la segunda pantalla. Para que se visualice GAmuza debe estar a Pantalla completa: Alt+f
- **USE VERTICAL SYNC:** sincroniza el número de frames con la tasa de refresco de la tarjeta gráfica, para reducir defectos visuales, como bandas o parpadeos y tener frames constantes



Bajo la ventana del previo de vídeo, hay un monitor de los fotogramas por segundo a los que está procesando GAmuza, si se selecciona Vertical Sync se puede observar el cambio.

TIMELINE

0 | LINE CODING (SECOND SCREEN) | 25 | LSCREEN OUTPUT @ 44 x 50 | 56 | 75 | 100 | 125 | 150 | 175

POSX

- DRAW MAPPING GRID
- RESET MAPPING POINTS
- LOAD MAPPING SETTING
- SAVE MAPPING SETTING
- COLOR CORRECTION SETTINGS
- GAMMA CORRECTION 1.00
- BRIGHTNESS 1.00

POSY

- CONTRAST 1.00
- FILM BLEACH 0.00
- FILM TECHNICOLOR 0.00
- FILM FORCE B&W 0.00

TRIGGERS

- FILM WHITE DIFFUSION 0.0
- VIDEO OUTPUT SETTINGS
- ACTIVATE SECONDARY SCREEN
- USE VERTICAL SYNC

FPS
MAX: 200
FPS: 145.98

26 | WHITE | 82 | PINK | 103 | WHITE | 148 | PINK

LINEAR
 SINE
 CIRCULAR
QUADRATIC
 CUBIC
 QUARTIC
 QUINTIC
 EXPONENTIAL
 BACK
 BOUNCE
 ELASTIC
EASE IN
 EASE OUT
 EASE IN-OUT

2.2.1. Timeline

El Timeline¹⁶ no es exactamente un módulo de aplicación GUI, sino una ampliación del Live coding para facilitar la programación, con la que se puede visualizar el tiempo, poner keyframes (fotogramas-clave) y triggers (etiquetas). Las pistas de keyframe se definen con código en el ScriptEditor, utilizando la función de GAmuza `ga.addTimelineKeyframes()`, que lleva 4 parámetros: "nombreVariable", "nombreVariable.xml" (genera un fichero que almacena los datos de esa línea del timeline), Valor mínimo de la variable, Valor máximo.

```
ga.addTimelineKeyframes("posX", "posX.xml", 0, OUTPUT_W)
```

Se pueden hacer tantas líneas de keyframe como funciones se pongan, en la imagen se han definido 2, cuyos nombres de variable son; posX y posY. La duración en frames se determina con la función `ga.setTimelineFrames()`. Una vez generada la línea, los keyframes se sitúan temporalmente clicando con el ratón sobre el frame deseado, se pueden desplazar con el ratón, o borrarlos. La transición entre un keyframe y otro se establece mediante unas reglas de interpolación que que se visualizan clicando con el botón derecho sobre el keyframe, mostrando las siguientes opciones:

- lineal
- curva senoide
- circular
- parábola de una función cuadrática
- cúbica
- quíntica (ecuación polinómica de quinto grado)
- exponencial
- técnicas de backoff
- con rebotes
- elástica

En todas ellas se puede añadir la característica:

- entrada suavizada
- salida suavizada
- entrada y salida suavizada



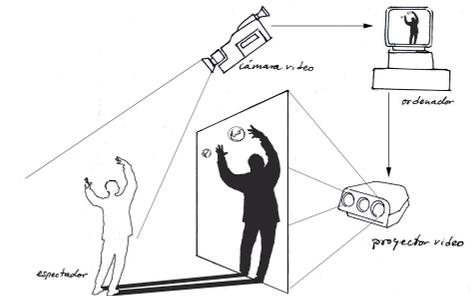
La pista inferior se denomina Trigger. Los trigger son elementos que activan un evento, actúan como etiquetas reconocibles, reduciendo mucho las líneas que deberían escribirse en el script si estos elementos, keyframes y triggers, no estuvieran.

La línea inferior del Timeline actúa como un escalímetro para ampliar/reducir su tamaño. También puede regularse la altura de las líneas de keyFrame con el ratón.

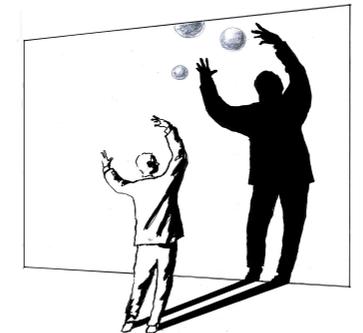
¹⁶ La base del TimeLine procede de James George, ofxTimeline <<http://jamesgeorge.org/opensource.html>> [12.08.2012]

2.3. Análisis Video

Los sistemas de tracking video están vinculados al campo de computer vision, su aplicación en el ámbito del arte proporciona una información importante para explorar nuevos modos de conexión e interactividad entre el cuerpo físico del espectador y la representación del mundo digital. A través del ojo de la cámara, el sistema reconoce posiciones, movimientos, colores, gestos, rostros, dentro de su campo visual, siempre que las condiciones de iluminación sean favorables.



Conociendo esos datos, se pueden programar aplicaciones ante las que el espectador interactúa de forma más intuitiva, denominándose por ello interfaces gestuales, porque la comunicación se produce sin tocar interfaces físicos. En el arte interactivo y electrónico son numerosos los ejemplos de este tipo de aplicaciones, que también ha incrementado su presencia en las artes escénicas.



Basado en *Bubbles*, (2002) W. Muench y K. Furukawa

Estos dos ejemplos mantienen una relación inversa entre obra y el espectador. En *Bubbles*¹⁷, una de las piezas pioneras de arte interactivo con tracking video, el espectador juega con las formas proyectadas, modificando su movimiento al colisionar con su sombra. En *Body Navigation*¹⁸, el cuerpo de los bailarines genera las formas. Aunque en ambos casos, es el sistema el que las modifica o produce al conocer con precisión la posición del cuerpo del espectador o de los bailarines.



Basado en *Body Navigation* (2008) O. Kristensen y J. Jongejan

El tracking de movimiento, colores o formas puede utilizarse también para generar o modificar sonidos, activar o desactivar dispositivos, simplemente hay que comprender su procedimiento.

Por ello, a continuación se describe la interface del módulo Computer Visión que cubriría las primeras fases de este proceso: analizar determinadas características de algo exterior, recoger datos de su comportamiento, seleccionar y/o ajustar esos datos, para después generar con ellos otra situación que responda en tiempo real a sus cambios.

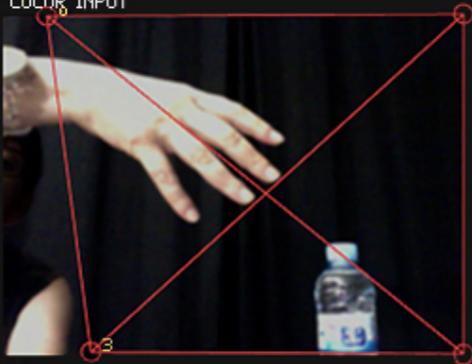
¹⁷ *Bubbles*, (2002) W. Muench y K. Furukawa <<http://at.zkm.de/node/193>> [10.07.2012]

¹⁸ *Body Navigation* (2008) O. Kristensen y J. Jongejan <<https://vimeo.com/1362832>> [10.07.2012]

GAMUZA [0.4.1] 'SILENT SHORE' BETA | <HTTP://WWW.GAMUZA.CC> | <HTTP://WWW.D3COD3.ORG> SAVE LOAD

CAM DEVICE [2] | Built-in iSight

COLOR INPUT



USE DEVICE

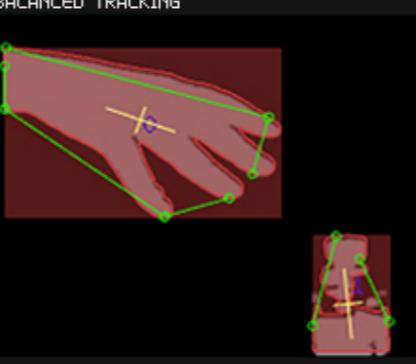
UNDISTORT [LENS CORRECTION]

MOTION DETECTION



RESET QUAD WARPING

BALANCED TRACKING



CAPTURE BACKGROUND

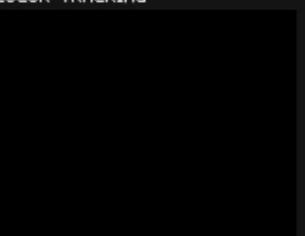
USE CONTRAST STRETCH

BACKGROUND SUBTRACTION



USE COLOR TRACKING

COLOR TRACKING



GENERAL SETTINGS

FLIP SOURCE IMAGE
HORIZONTAL

SOURCE BLUR 2

CONTRAST 0.00

BRIGHTNESS 0.00

BACKGROUND SUBTRACTION SETTINGS

SUBTRACTION TECHNIQUE
DARKER THAN

SUBTRACTION THRESHOLD 29

BLUR 0

ERODE 0

DILATE 0

OSC DATA SETTINGS

USE KALMAN FILTER

SMOOTHING FACTOR 0.870

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 0

DILATE 0

MOTION DETECTION SETTINGS

MOTION THRESHOLD 100

MOTION NOISE COMPENSATION 483

MOTION TRIGGER LOW LIMIT 20

MOTION TRIGGER RANGE LIMIT 50

BLOB TRACKING SETTINGS

MIN BLOB 1029

MAX BLOB 63333

CONTOUR DETAIL
SMOOTH

CONTOUR SMOOTH FACTOR 0.549

CONTOUR SIMPLE TOLERANCE 16.1

BALANCED TRACKING SETTINGS

BALANCED TRACKING BLUR 5

ERODE 1

DILATE 1

COMPUTING ALGORITHM SELECTOR

COMPUTE CONTOUR FINDER

COMPUTE CONTOUR GEOMETRY

COMPUTE OPTICAL FLOW

COMPUTE HAAR FINDER

COMPUTE TRIGGER AREAS

2.3.1. Computer vision

El módulo Computer vision¹⁹ se corresponde con la pestaña roja y facilita la aplicación de algoritmos para tracking video, permitiendo regular con sliders, menús y botones las condiciones básicas de entrada de datos al sistema. Como en el apartado Live coding, pasamos a describir su interface.

El monitor de entrada de la señal video, **COLOR INPUT**, permite reducir el área a trackear, este *crop* se ajusta arrastrando con el ratón los puntos resaltados de los vértices. Los demás monitores muestran solo la zona seleccionada. El crop se resetea con **RESET QUAD WARPING**.

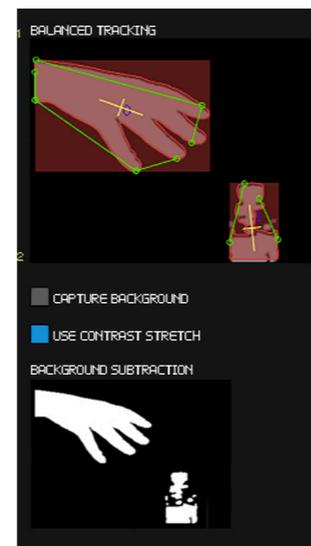
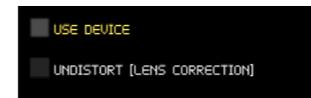
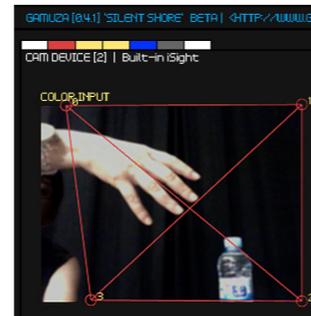
USE DEVICE. Activa/ desactiva el módulo, cuando no se utiliza pero sí se usará después, sin tener que reiniciar el programa.

UNDISTORT (LENS CORRECTION), sirve para desactivar ajustes previos hechos con la aplicación Lens Correction²⁰.

El monitor **BALANCED TRACKING** muestra el resultado final de todos los algoritmos de tracking aplicados y ajustados con los sliders, menús y botones del bloque derecho.

CAPTURE BACKGROUND, al clicar el botón se captura un frame de la imagen video que debe corresponder con el fondo. Esta imagen se toma como referencia para comparar los cambios que se producen entre el valor de sus píxeles y el de los frames de la imagen a trackear, aplicando el algoritmo Background Subtraction. Es el primer nivel para el cálculo de Tracking, y siempre activo. Según las condiciones de iluminación, puede mejorarse la imagen activando **USE CONTRAST STRETCH**: incrementa el rango de los valores más intensos de la imagen en escala de grises.

El resultado de **BACKGROUND SUBTRACTION** se ve en el monitor inferior.



¹⁹ Las aplicaciones de tracking video se han articulado en dos interfaces diferentes por las particularidades técnicas del sensor kinect, por ello en este módulo nos referiremos al uso de cualquier webCam o cámara de vídeo conectada y que el ordenador reconozca

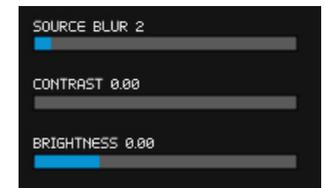
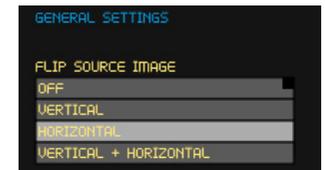
²⁰ Lens Correction es una aplicación, disponible en la web de GAmuza, diseñada para corregir las aberraciones de las ópticas.

GENERAL SETTINGS

FLIP SOURCE IMAGE voltea la imagen-video que proviene de la cámara, su ajuste depende de las condiciones de exhibición:

- **OFF**: la imagen se percibe sin ningún cambio
- **HORIZONTAL**: se selecciona cuando los movimientos ante la imagen responden al efecto de estar frente a un espejo.
- **VERTICAL**: imagen video cabeza abajo.
- **HORIZONTAL + VERTICAL**: voltea en los dos sentidos.

SOURCE BLUR: desenfoque de la imagen de entrada de video. **CONTRAST**: contraste. **BRIGHTNESS**: brillo.



BACKGROUND SUBTRACTION SETTINGS

SUBTRACTION TECHNIQUE, seleccionar entre:

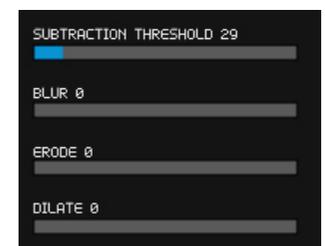
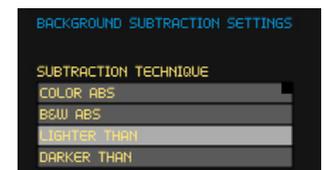
- **COLOR ABS**: color absoluto
- **B&W ABS**: blanco y negro absoluto
- **LIGHTER THAN**. Cuando la figura a trackear es más clara que el fondo.
- **DARKER THAN**. Cuando La figura a trackear es más oscura que el fondo.

SUBTRACTION THRESHOLD, ajusta el umbral de detección.

BLUR, desenfoca la imagen procesada

ERODE, reduce las superficies detectadas

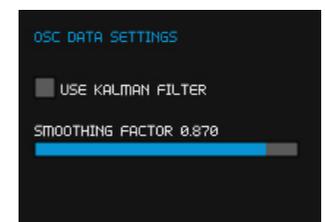
DILATE, expande las superficies detectadas



OSC DATA SETTINGS

USE KALMAN FILTER. En la transferencia de datos a otras aplicaciones, el filtro de Kalman suaviza los cambios grandes entre los valores obtenidos, desechando aquellos que superan unos límites establecidos.

SMOOTHING FACTOR ajustar el rango de suavizado.



USE COLOR TRACKING

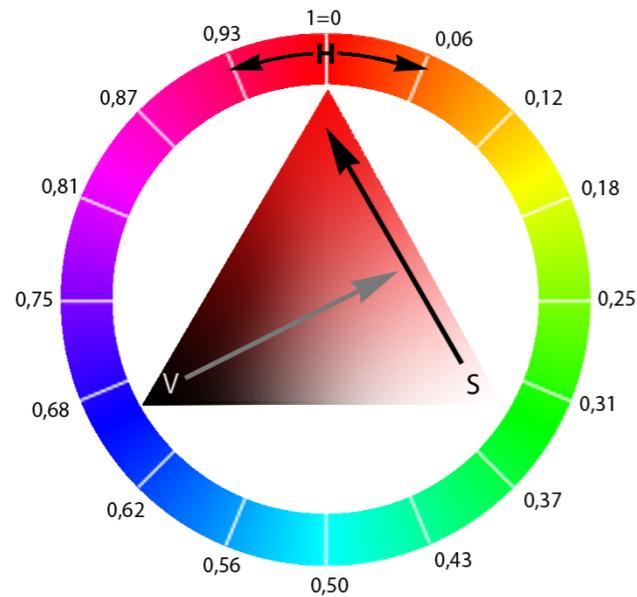
Color Tracking es un algoritmo de reconocimiento del color que requiere condiciones de iluminación estables y la elección de un color de seguimiento muy diferenciado del contexto.

USE COLOR TRACKING, activa/desactiva su uso y los resultados se ven en el monitor inferior.



COLOR TRACKING SETTINGS

La configuración de este bloque solo se ajustan si está activado **USE COLOR TRACKING**. El Tracking Color transforma el espacio de color RGB al sistema HSV, porque asigna los valores del tono del color de una forma más estable. HSV identifica cada tono con un número que va de 0° a 360° según su posición en el círculo cromático; en GAmuza el rango va de 0 a 1 quedando su equivalencia tal como muestra la imagen.



Las opciones de ajuste son:

HUE, selecciona el tono a trackear según la escala de valores de GAmuza entre 0 y 1. La franja de color sirve de orientación.

HUE RANGE, amplía el rango del tono seleccionado.

SATURATION, asigna el nivel de saturación del tono (hue) seleccionado. El valor 0 es igual a blanco, 1 es el tono puro, dependiendo del nivel de luminosidad (value).

SATURATION RANGE, amplía el rango de saturación seleccionado.

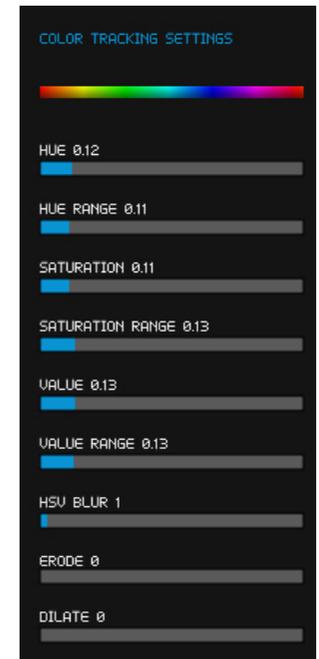
VALUE, regula el nivel de luminosidad (value) del tono. El valor 0 es igual a negro, 1 es igual al tono puro, dependiendo del valor de saturación.

VALUE RANGE, amplía el rango de luminosidad seleccionado.

HSV BLUR, filtro de desenfoque de la imagen en HSV.

ERODE, reduce las superficies detectadas.

DILATE, expande las superficies detectadas.



BLOB TRACKING SETTINGS

Los blob son las zonas independientes que el tracking detecta según los algoritmos seleccionados, sus opciones de configuración son:

MIN BLOB, tamaño mínimo de los blobs a detectar.

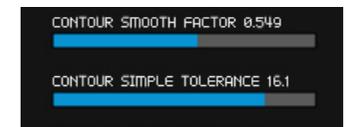
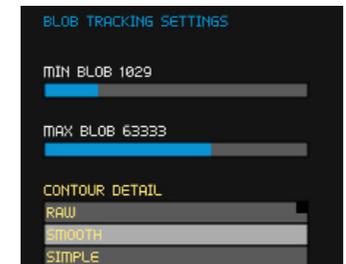
MAX BLOB, tamaño máximo de los blobs a detectar.

CONTOUR DETAILL, el detalle del contorno se puede ajusta en:

- **RAW**, todo el perímetro tal cual es
- **SMOOTH**, suavizado
- **SIMPLE**, adaptado a formas geométricas simples

CONTOUR SMOOTH FACTOR, si se ha seleccionado smooth, ajusta el nivel de suavizado

CONTOUR SIMPLE TOLERANCE, si se ha seleccionado simple, ajusta el rango de vértices que articulan el contorno.

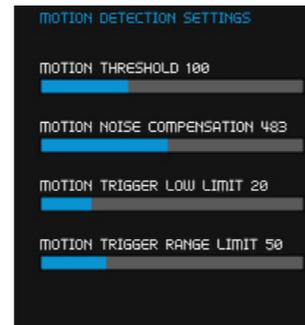


MOTION DETECTION SETTINGS

El algoritmo para detección de movimiento está siempre activo. El monitor **MOTION DETECTION** muestra el cálculo de la cantidad de movimiento que se registra en la imagen y señala la posición media de ese movimiento. Los ajustes para su configuración son:



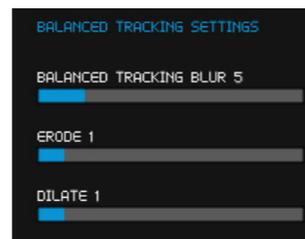
- **MOTION THRESHOLD**, regula la sensibilidad del sistema para activar la detección, si es poco sensible se debe ampliar el umbral (Threshold)
- **MOTION NOISE COMPENSATION**, compensa el ruido de la señal de imagen video
- **MOTION TRIGGER LOW LIMIT**, nivel más bajo para activar la detección de movimiento
- **MOTION TRIGGER RANGE LIMIT**, limite del rango para la detección de movimiento.



BALANCED TRACKING SETTINGS

Vuelve a ajustar, en conjunto, los valores dados en cada sección.

- **BALANCED TRACKING BLUR**, ajusta el nivel de desenfoque global,
- **ERODE**, reduce las superficies detectadas
- **DILATE**, expande las superficies detectadas



COMPUTING ALGORITHM SELECTOR

En este bloque se seleccionan otros algoritmos específicos que se pueden aplicar, además de los mencionados hasta ahora.

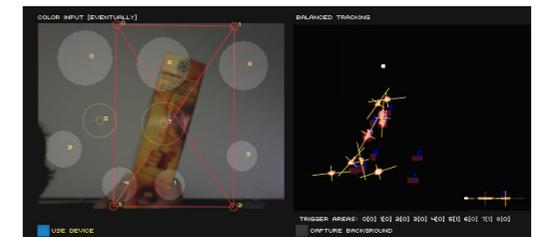
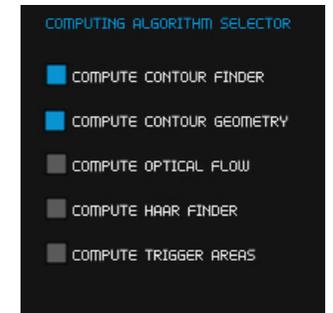
COMPUTE CONTOUR FINDER, para procesar el contorno de las zonas detectadas. Sus valores se ajustan con **BLOB TRACKING SETTINGS**.

COMPUTE CONTOUR GEOMETRY. Depende directamente del contour finder,

COMPUTE OPTICAL FLOW, calcula el flujo óptico del movimiento y lo describe con vectores que indican la dirección e intensidad.

COMPUTE HAAR FINDER, activa el algoritmo para detectar partes del cuerpo humano. Hay distintos algoritmos posibles y debe seleccionarse previamente en el Setting Manager (ver página XXX). Por defecto está seleccionado **FRONTALFACE_ALT**.

COMPUTE TRIGGER AREAS. Activa la posibilidad de utilizar 8 zonas circulares, que están numeradas para su identificación. Su posición y tamaño pueden ajustarse con el ratón; con ellas se pueden determinar las áreas donde responde el tracking utilizando las funciones específicas de GAmuza para ello (ver página XXX)



Si hay varias cámaras conectadas al ordenador, habrá un interface Computer Visión para cada una de ellas, es decir, si hay dos cámaras conectadas al ordenador, habrán dos pestañas rojas, cada una abre una interface diferente que puede ajustarse independientemente.

2.3.2. Sensor Kinect

GAMUZA [0.4.1] "SILENT SHORE" BETA | <HTTP://WWW.GAMUZA.CC> | <HTTP://WWW.D3C003.ORG> SAVE LOAD

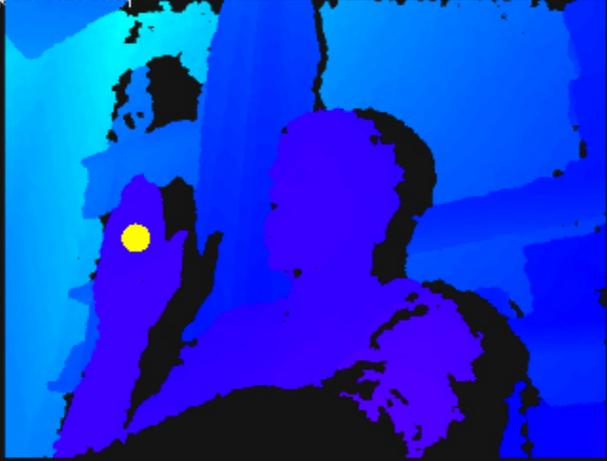
OPENNI SENSOR KINECT DEVICE

RECORD IMAGE



COMPUTE SENSOR KINECT

RECORD DEPTH



NEAR TRESHOLD 408

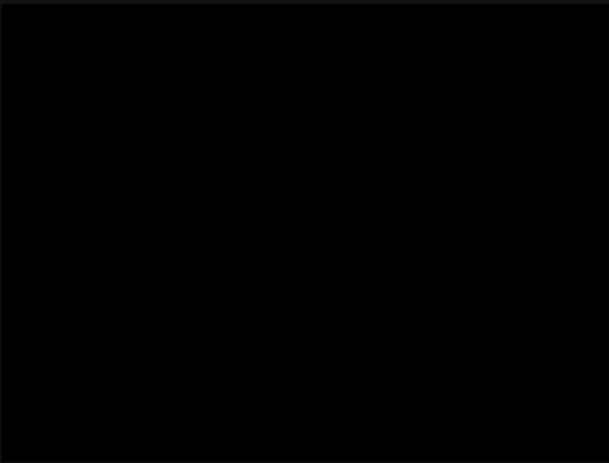
FAR TRESHOLD 1122

DEPTH RANGE MASK



USE KINECT HANDS TRACKING

ALL USERS MASK



GENERAL SETTINGS

SENSOR KINECT BLUR 2

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: 0.07

MIN: 0

ACCELEROMETER X MAX: 1

X: 0.48

MIN: 0

ACCELEROMETER Y MAX: 1

Y: 0.92

MIN: 0

ACCELEROMETER Z MAX: 1

Z: 0.72

MIN: 0

La configuración del módulo kinect mantiene muchas semejanzas con el de computer visión, excepto en las particularidades del sensor.

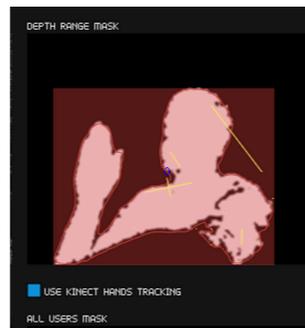
RECORD DEPTH: Los dos sliders bajo el monitor permiten ajustar y visualizar el rango de profundidad a detectar:

- **NEAR TRESHOLD:** determina el plano de los elementos más próximos, como mínimo desde 70 cm del sensor.
- **FAR TRESHOLD:** hasta qué plano de profundidad se quiere detectar, como máximo 6 m.



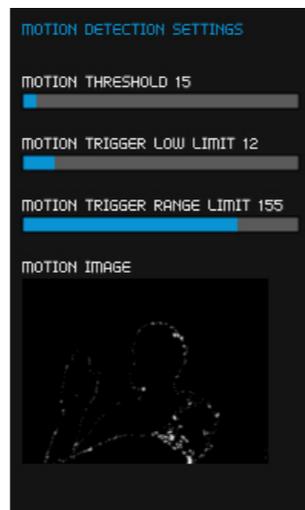
DEPTH RANGE MASK: monitor que muestra el resultado de los algoritmos aplicados.

USE KINECT HANDS TRACKING: activa el sistema de detección de manos, se representa en el monitor de la imagen anterior con puntos amarillos.



MOTION DETECTION SETTINGS: puede regular la detección de movimiento según niveles de profundidad.

- **MOTION TRESHOLD:** regula la sensibilidad del sistema para activar la detección, si es poco sensible se debe ampliar el umbral (Threshold)
- **MOTION TRIGGER LOW LIMIT:** nivel más bajo para activar la detección de movimiento
- **MOTION TRIGGER RANGE LIMIT,** limite del rango para la detección de movimiento.



GENERAL SETTINGS

SENSOR KINECT BLUR: grado de desenfoque de la señal de entrada

ERODE: reducir el área de las superficies detectadas

DILATE: ampliar el área de las superficies detectadas

BLOB TRACKING SETTINGS

MIN BLOB: tamaño mínimo de los blobs

MAX BLOB: tamaño máximo de los blobs

CONTOUR DETAIL: menú con tres opciones, raw, smooth o simple

CONTOUR SMOOTH FACTOR, si se ha seleccionado smooth, ajusta el nivel de suavizado

CONTOUR SIMPLE TOLERANCE, si se ha seleccionado simple, ajusta el rango de vértices que articulan el contorno.

COMPUTING ALGORITHM SELECTOR

COMPUTE CONTOUR FINDER, para procesar el contorno de las zonas detectadas. Los valores se ajustan con **BLOB TRACKING SETTINGS.**

COMPUTE CONTOUR GEOMETRY, determina formas geométricas a partir de los contornos. Al activarlo se activa automáticamente el anterior si no lo estaba.

COMPUTE OPTICAL FLOW, calcula el flujo óptico del movimiento y lo describe con vectores que indican la dirección e intensidad.

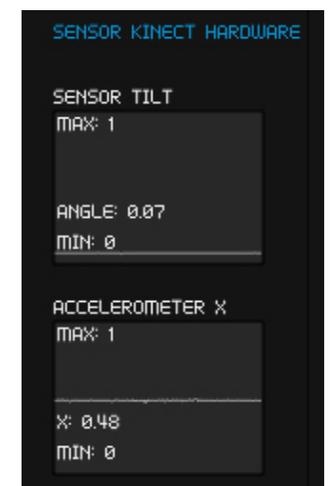
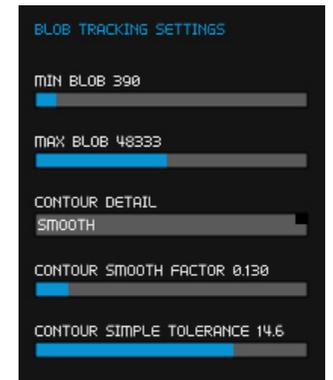
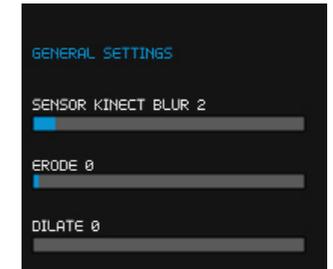
COMPUTE TRIGGER AREAS. Activa la posibilidad de utilizar 8 zonas circulares, como en el módulo computer vision.

SENSOR KINECT HARDWARE

Recoge datos de los sensores internos de la Kinect

SENSOR TILT: datos de inclinación

ACCELEROMETER X, Y, Z: datos independientes del acelerómetro en los ejes x, y y z



2.4. Análisis audio

Lo natural es que todos los sentidos reciban información. Lo antinatural es el aislamiento de los sentidos.²¹

El sonido es un medio pasajero que se mueve en el tiempo, difícil de asir, pero la incorporación del audio en los medios digitales como un elemento casi propio, ha modificado nuestra relación con él, cambiando los hábitos de escucha y percepción del sonido. Sin perder su cualidad temporal se ha incrementado mucho su capacidad espacial. En el campo del arte estos cambios se ven reflejados en las instalaciones de arte sonoro o en las actuaciones de música con visuales, eventos que en muchos casos han ido borrando la distancia que los separaba, al incorporar sonido espacializado o propuestas sinestésicas entre audio, vídeo, luz y/o gráficos, estableciendo conexiones directas o indirectas entre la materialidad de lo sonoro con la sonoridad de lo visual. El propio sistema Live coding en que se basa GAmuza es consecuencia o reflejo de estos cambios.

Muchas de estas propuestas artísticas requieren el análisis de los parámetros del sonido. Para el sistema estos datos suponen solo valores numéricos que pueden aplicarse para producir otras situaciones sonoras, visuales o físicas, y es la casi simultaneidad de los cambios que se producen en el sonido analizado y en la respuesta generada la establece un efecto empático entre ambas.

Métodos de medición se convierten en observaciones físicas. Los oídos se vuelven instrumentos ópticos; el movimiento de los ojos, una manera de entender el funcionamiento del sonido.²²



Basado en la obra test pattern [nº3] (2010) de Rioji Ikeda²³

Pasamos a describir la interface Audio Stream de GAmuza que analiza la entrada de sonido, obteniendo datos del pitch, volumen o FFT, y realizar otros ajustes para el tratamiento de la señal de audio.

²¹ José Manuel Costa (2011), "Arte sonoro" en Exit Express nº 54, pág. 26

²² Magnus Haglund (2005) "The air between the planets, on the audio and visual works of Carsten Nicolai", en Carsten Nicolai Anti Réflex, Frankfurt: Max Hollein, pág 27.

²³ Rioji Ikeda < <http://www.ryojiikeda.com/project/testpattern/> > [08.09.2012]

AUDIO INPUT [CHANNEL 0]

GENERAL SETTINGS

USE INPUT CHANNEL

VOLUME 1.000

MUTE

NOISE REDUCTION SETTINGS

REC NOISE

REDUX FACTOR 2.50

PARAMETRIC EQUALIZATION

BAND 1 CENTER [BIN] 35.6

BAND 1 AMPLITUDE [DB] 1.45

BAND 1 WIDTH [Q] 8.5

BAND 2 CENTER [BIN] 108.1

BAND 2 AMPLITUDE [DB] -1.30

BAND 2 WIDTH [Q] 9.3

BAND 3 CENTER [BIN] 197.7

BAND 3 AMPLITUDE [DB] 1.90

BAND 3 WIDTH [Q] 7.8

OSC DATA SETTINGS

USE KALMAN FILTER

SMOOTHING FACTOR 0.413

VOLUME CHART
MAX: 1

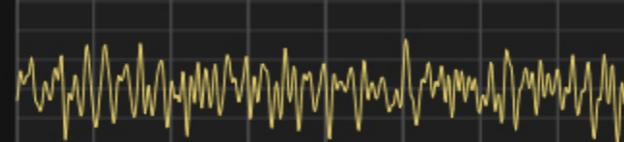
CHANNEL VOLUME: 0.11
MIN: 0

PITCH CHART
MAX: 20000

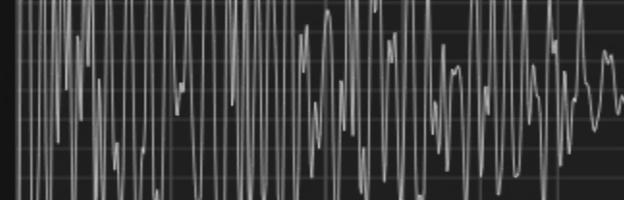
CHANNEL PITCH: 69.63
MIN: 20

TIME DOMAIN

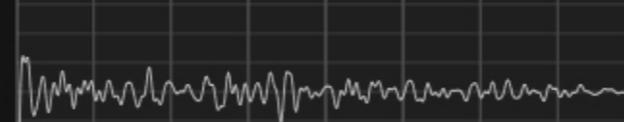
RAW SIGNAL



CORRELATED SIGNAL



CORRELATED SIGNAL NORMALIZED



FILTERED CORRELATED SIGNAL



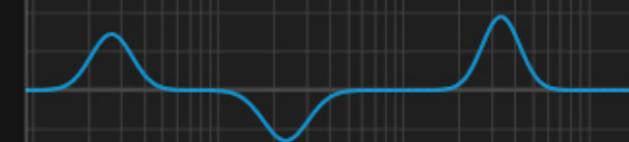
TIME (S)

FREQUENCY DOMAIN

NOISE FILTER



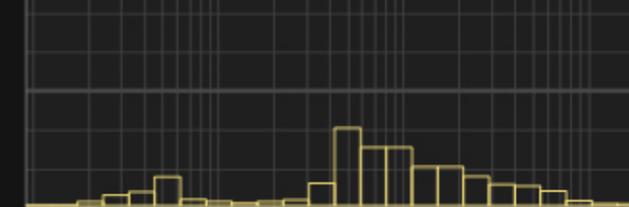
PARAMETRIC EQ [NORMALIZED RADIAL BASIS FUNCTION NETWORK]



FILTERED FFT BINS [257]



BARK SCALE BINS [24]



FREQUENCY (HZ)

2.4.1. Audio Stream (Input/Output)

Antes de describir las opciones de la interface, señalar que este módulo GUI es solo para ajustar el análisis de entrada de sonido (Input Channel), no la salida. Si en el Setting Manager se ha activado el módulo Audio Stream (input/output) para generar sonido, y se ha seleccionado un dispositivo solo de salida, esta interface no estará visible, pero GAmuza mantendrá comunicación con los dispositivos de reproducción de audio. Los portátiles mac suelen tener dispositivos separados para entrada y salida, por lo que es común esta situación, la mayoría de las tarjetas de audio, por el contrario, permiten ambas cosas a la vez.

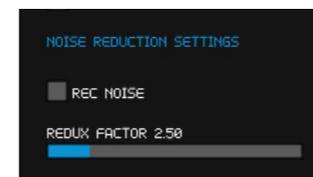
Si el dispositivo seleccionado tiene varios canales de entrada, veremos varios interfaces de Audio Input iguales con ajustes independientes, a cada uno de ellos se accede clicando en una de las pestañas amarillas. Debajo de las pestañas se indica el canal que estamos visualizando [CHANNEL 0], [1], [2]..., que será necesario poner como parámetro en muchas de las funciones de programación de audio.



Si sólo se va a analizar la entrada de un canal, se pueden desactivar los otros con el botón USE INPUT CHANNEL.

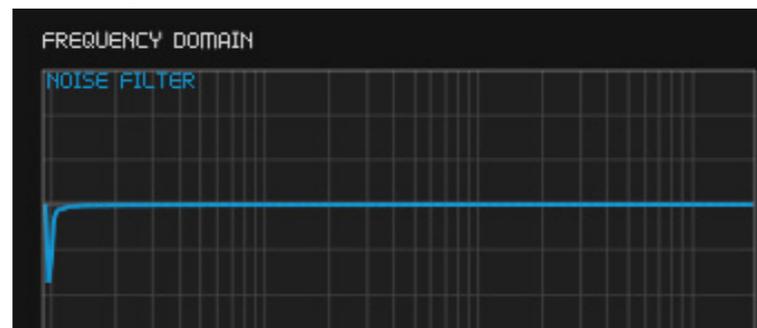
Debajo hay un slider para regular el nivel de volumen y un botón para silenciarlo.

NOISE REDUCTION SETTINGS es un sistema para grabar el sonido ambiente que se desee reducir. Clicando REC NOISE, se inicia la grabación y clicando de nuevo se detiene.

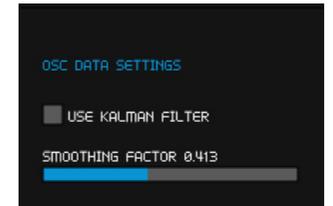


El slider, REDUX FACTOR, gradúa el nivel de reducción a aplicar.

Este proceso se puede visular en el gráfico NOISE FILTER, tanto la grabación como el ajuste de reducción posterior.



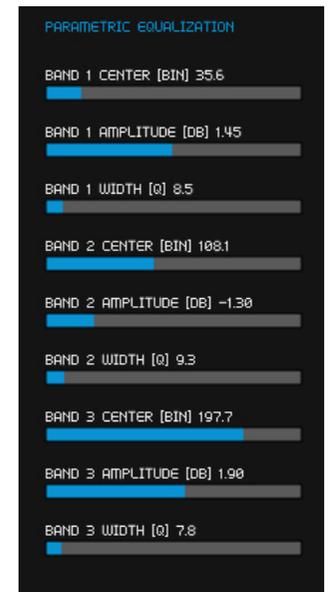
Como en los otros módulos GUI, el envío de los datos de análisis de audio por OSC puede ajustarse activando USE KALMAN FILTER. El filtro de Kalman suaviza la relación de los datos cuando presentan grandes cambios, cortando aquellos que superan unos límites establecidos. SMOOTHING FACTOR ajusta el rango de ese suavizado.



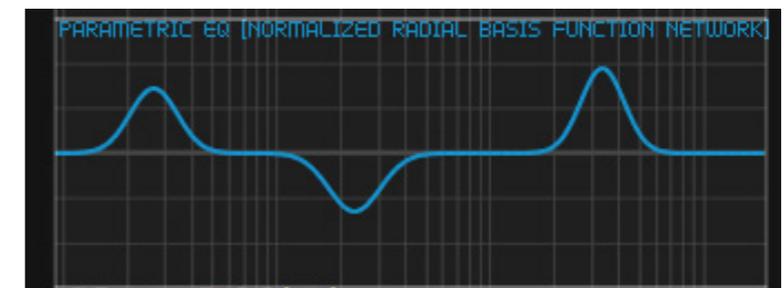
El siguiente bloque de ajustes, PARAMETRIC EQUALIZATION, equaliza la señal de audio, ajustando el control individual de tres parámetros, en tres bandas: frecuencia central, ganancia y ancho de banda.

- **CENTER [BIN]**, selecciona la frecuencia central en Herzios.
- **AMPLITUDE [DB]**, controla la ganancia, determinando cuánto se amplifican o recortan esas frecuencias, por eso puede tener valores negativos, como muestra la imagen y el gráfico en la segunda banda.
- **WIDTH [Q]**, el factor Q estrecha o amplía el rango de frecuencias seleccionadas alrededor de CENTER [BIN].

Los ajustes se van visualizando en el gráfico PARAMETRIC EQ. El gráfico siguiente se corresponde a los valores ajustados en los sliders de la imagen derecha. Primero se busca la posición de la frecuencia central y se le ajustan la ganancia y el factor Q.



NORMALIZED RADIAL BASIS FUNCTION NETWORKS (RBFN) es un algoritmo que genera un sistema de interpolación de datos para representar el filtro que se está aplicando a la señal de audio, en el dominio de la frecuencia.



VOLUME CHART muestra el gráfico y valores del volumen. Su rango de valores va de 0.0 a 1.0. La visualización de estos datos permite calcular mejor si deben escalarse, y por cuánto, para su uso en programación. La función de GAmuza que recoge el volumen es `ga.getVolume()`.

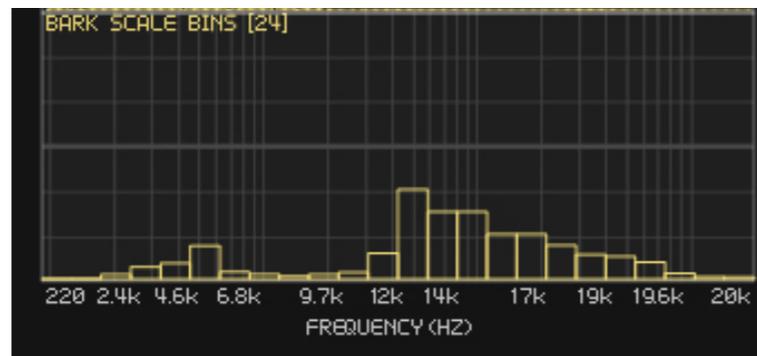


PITCH CHART muestra igualmente el gráfico y valores del tono (pitch), su rango oscila entre 20Hz to 20000Hz, pero cuando los recoge la función de GAmuza, `ga.getPitch()`, los normaliza entre 0.0 y 1.0

Hay otros dos datos que GAmuza obtiene de este módulo: BIN y FFT.

Los Bins provienen de la escala de Bark²⁴, una escala psicoacústica propuesta por Eberhard Zwicker en 1961, que corresponde a la subdivisión de las primeras 24 bandas críticas del oído. Los márgenes de las bandas en Herzios son 0, 100, 200, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500.

GAmuza obtiene el valor de cada Bin desde el análisis del FFT. Para trabajar con estos datos se utiliza la función `ga.getBin()` y la variable interna `FFT_BANDS`.

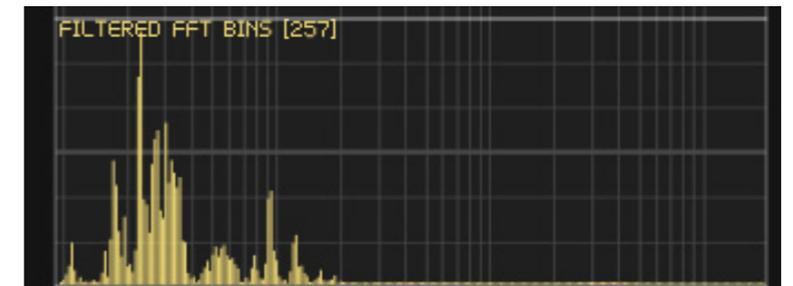


La transformada rápida de Fourier, FFT (Fast Fourier Transform), es un algoritmo muy utilizado para el tratamiento de la señal de audio. En GAmuza opera sobre las muestras de sonido almacenadas en el BufferSize, por lo que su rango de valores depende del tamaño de buffer que se haya asignado en el SettingManager.

Si hay un tamaño de buffer 512, el gráfico Filtered FFT BINS tendrá un tamaño [257], si el buffer es 1024, ese valor cambia a [513], por lo que el algoritmo establece que el tamaño del FFT es el del BufferSize /2 +1, que serán los puntos de análisis de la frecuencia.

En programación, estos datos se utilizan con la función `ga.getFFT()` que tiene dos parámetros: ID del canal de entrada de audio: [CHANNEL num], y la posición que ocupa la muestra (sample) en el Buffer, recorriéndolo desde 0 al valor de la variable global interna `BUFFER_SIZE`. Ejemplo:

```
for i=0,BUFFER_SIZE-1 do
  of.box(ga.getFFT(0,i)*200)
end
```



²⁴ Definición Bark scale [enciclopedia on-line] http://en.wikipedia.org/wiki/Bark_scale [25.08.2012]

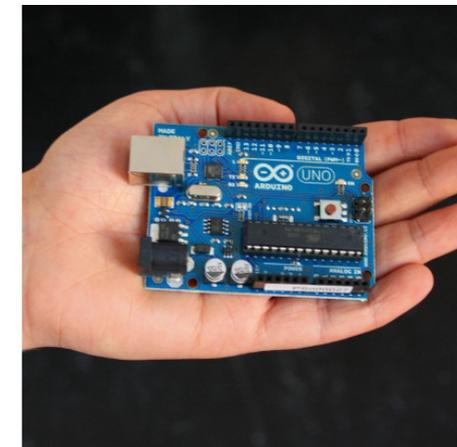
2.5. Interfaces físicas

Cuando muchos de los estudios sobre arte digital auguraban un imparable crecimiento de lo virtual frente a lo físico, empezó a surgir toda una línea de trabajos en el campo del arte electrónico y el diseño que hibridaban lo físico y lo virtual, desarrollando interfaces físicas, con un carácter muy diferente al teclado o el ratón, para comunicarse con el ordenador o para comunicar los datos que genera o recoge el ordenador con otros dispositivos electro-mecánicos o electrónicos.

El ordenador no solo puede recoger datos a través de la cámara de vídeo o el análisis de sonido, también mediante sensores -que detectan magnitudes físicas o químicas y las transforman en variables eléctricas-, un microcontrolador que haga de intérprete y un software, puede recoger tantos tipos de datos como los que registran los sensores²⁵. El microcontrolador más utilizado en el campo del arte electrónico para este tipo de trabajos es Arduino.

Arduino es una plataforma de open hardware, basada en una placa con un microcontrolador y un entorno de desarrollo open source, diseñada para facilitar el uso de electrónica en proyectos multidisciplinarios.

hardware abierto significa tener la posibilidad de mirar lo que hay dentro de las cosas, que eso sea éticamente correcto, y que permita mejorar la educación. Educar en cómo funcionan las cosas...²⁶



²⁵ Tipos de sensores *enciclopedia on-line* <http://es.wikipedia.org/wiki/Sensor#Tipos_de_sensores> [30-08.2012]

²⁶ David Cuartielles (2010) *Arduino The Documentary*, [video on-line] <<https://vimeo.com/18390711>> [30-08.2012]

GAMUZA [0.4.1] 'SILENT SHORE' BETA | <HTTP://WWW.GAMUZA.CC> | <HTTP://WWW.D3C0D3.ORG> SAVE LOAD

ARDUINO DEVICE CONNECTED AT SERIAL PORT /dev/ttyBluetooth-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: 0.00
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 OFF

PIN 1
REPORT OFF

PIN 2
REPORT OFF

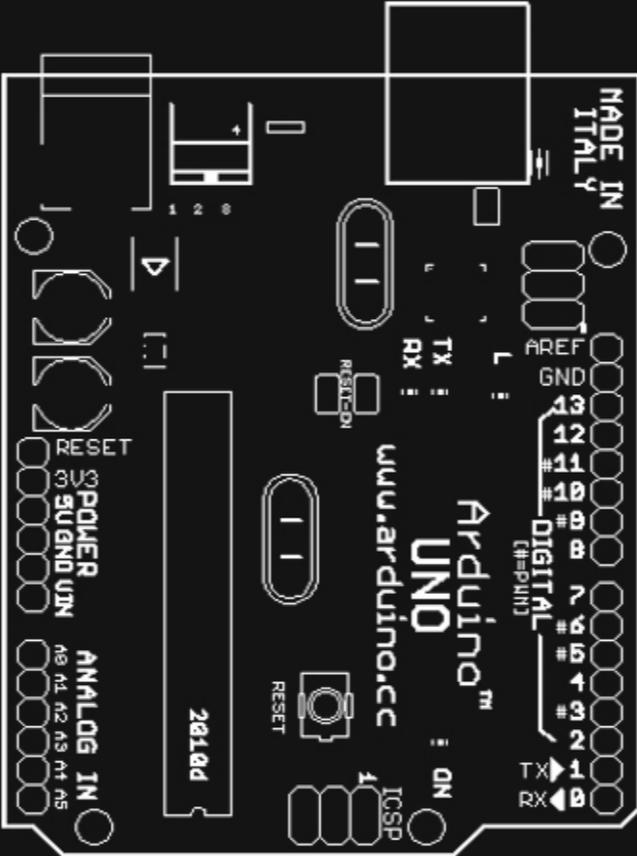
PIN 3
REPORT OFF

PIN 4
REPORT OFF

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
PWM

PIN [PWM] 11
PWM

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] 100

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] 100

PIN 11 PWM [0 254] 122

PIN 12 LOW | HIGH

PIN 13 LOW | HIGH

2.5.1. Módulo Arduino

El módulo Arduino responde a la pestaña azul. Esta interface establece la comunicación entre GAmuza y el microcontrolador Arduino²⁷, las opciones de configuración son:

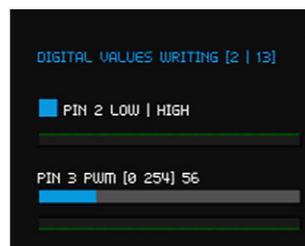
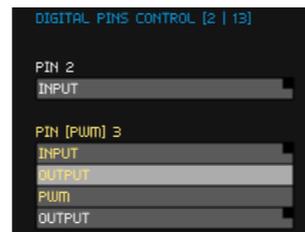
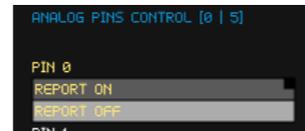
Lectura de valores analógicos, del PIN 0 al 5. Cada monitor muestra el gráfico de la señal y su valor entre 0 y 1023.

La función de GAmuza, `ga.analogRead()` normaliza ese rango de valores de 0.0 a 1.0.

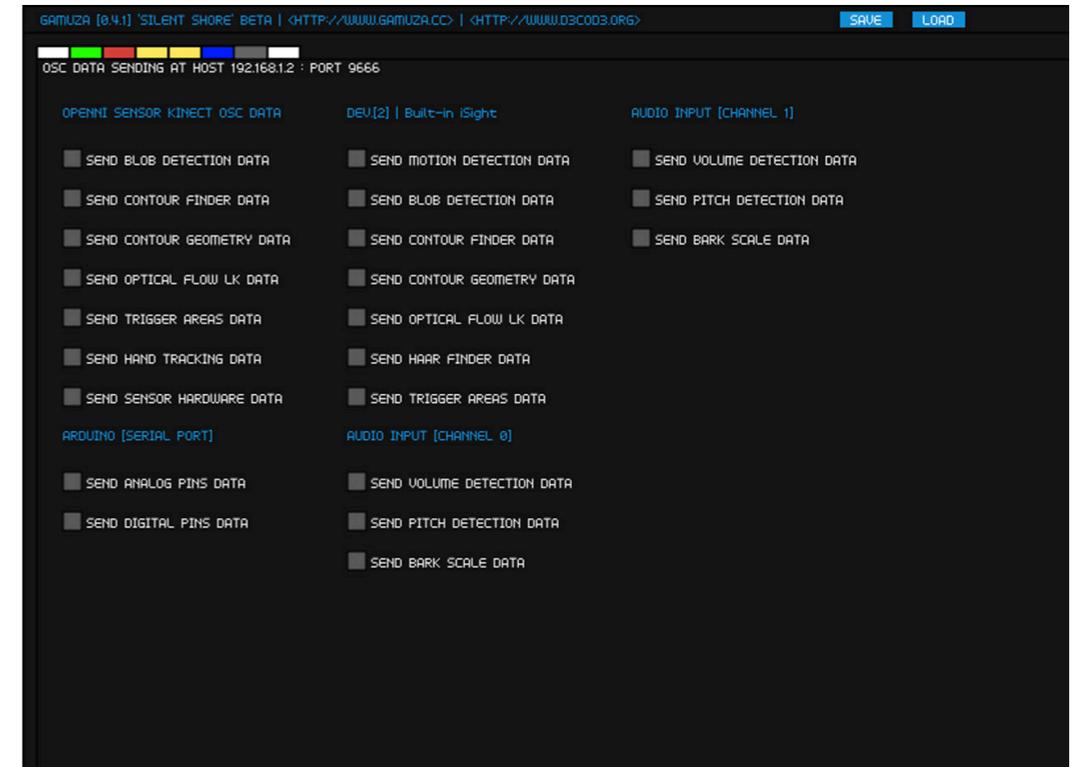
En **ANALOG PIN CONTROL**, hay que seleccionar **REPORT ON** en los pin que se deseen leer.

Arduino Uno²⁸ tiene 14 pin digitales, de los cuales 6 PWM (Pulse-width modulation) de 8 bits pudiendo transmitir un rango de valores de 0 a 254. Si se selecciona la opción Input u output funcionan como los demás, solo envían o reciben dos datos: HIGH o LOW.

DIGITAL VALUES WRITING: los ajustes de los valores de escritura de los pin digitales y PWM, están vinculados a los datos programados con la función de GAmuza `ga.digitalWrite(pin, valor)`, el pin al que haga referencia la función debe estar activado en la interface.



2.6. Comunicación OSC



La interface de Comunicación OSC está siempre presente (no se selecciona desde Setting Manager) y responde a la pestaña de color gris.

Su contenido varía en función de los módulos GUI que estén activados, estableciendo en su interface grupos de datos de cada uno de ellos, que presentan la opción de activar/desactivar el envío a otros programas de los datos individualizados que genera cada módulo.

El sistema de recepción de datos por OSC se hace mediante código, no en la interface, y se recibe por el puerto indicado en el Setting Manager. Ver ejemplos en `File/examples/Protocols/oscReceiving`, o `File/examples/Protocols/oscSending`.

²⁷ Arduino < <http://www.arduino.cc/> > [25.08.2012]

²⁸ Arduino Uno < <http://arduino.cc/en/Main/ArduinoBoardUno> > [06.09.2012]