

Software Open Source y el desarrollo de Nuevas Interfaces en el Arte Interactivo

Autores: Grupo Proyecto Biopus – (www.biopus.com.ar)

Emiliano Causa (e_causa@ec.gba.gov.ar), Tarcisio Pirotta (tarcisiopirotta@yahoo.com.ar) y

Matías Romero Costas (matiasromeroc@yahoo.com.ar).

Palabras claves:

Open Source, nuevas interfaces, captura de movimiento, realidad aumentada, captura de patrones.

1. Introducción

El desarrollo de nuevas interfaces (que exceden a las convencionales: teclado, mouse, monitor) en el arte interactivo viene creciendo desde hace aproximadamente una década. Nos encontramos, cada vez más, en presencia de instalaciones, intervenciones y performance donde se controla video y/o sonido en tiempo real, o ejecuta actuadores robóticos, en función del comportamiento del público o el performer. Se puede citar los trabajos de Rafael Lozano Hemmer, Marcel Li Antunez, Ken Rinaldo, Eduardo Kac, Naoko Tosa, los grupos Flong, Mine Control, Art+Com, Christa Sommerer y Laurent Mignonneau, por nombrar sólo los más destacados de una extensa lista.

En paralelo con este avance de los formatos expresivos, ha crecido la oferta de tecnologías más baratas y disponibles a todo público para este tipo de trabajos. Hace poco más de 5 años, las herramientas disponibles en el mercado para estos desarrollos tenía licencias pagas, de hecho, la herramienta de referencia era MAX, un software para composición algorítmica que luego devino en herramienta para control de sonido y video en tiempo-real (esto último con el paquete Jitter). Hoy en día, en cambio, se dispone de una gran variedad de software “open source” (código abierto), de gran especificidad, de los cuales muchos superan en varios aspectos a MAX y están soportados por comunidades enteras que los desarrollan, mantienen y documentan, a la vez que permiten a los usuarios participar de su desarrollo. Este cambio en los modos de producción y distribución del software se ve acompañado por uno equivalente en el hardware, en donde se dispone de una amplia gama de sensores, actuadores y placas digitalizadores (sumado al abaratamiento de la cámaras Web), que permiten desarrollar diversos sistemas de captación de movimiento, localización en el espacio, captación de patrones. Si hace media década atrás, este tipo de trabajos artísticos requerían pagar costosas licencias y disponer de dispositivos de precios privativos, hoy en día con

un hardware hogareño y software libre disponible en Internet, es posible construir trabajos de nivel profesional y que puede superar a las tecnologías anteriores.

Dada la extensión de este texto veremos sólo algunas de estas tecnologías.

2. Ambientes de programación visual para control y generación de sonido y video en tiempo real

Comencemos por ver el desarrollo de software “open source” que siguen la línea del conjunto MAX/Jitter. Este tipo de ambientes permite construir un programa (que aquí se le llamará “patch”) mediante objetos que podemos interconectar gráficamente para definir el tránsito del flujo de datos.

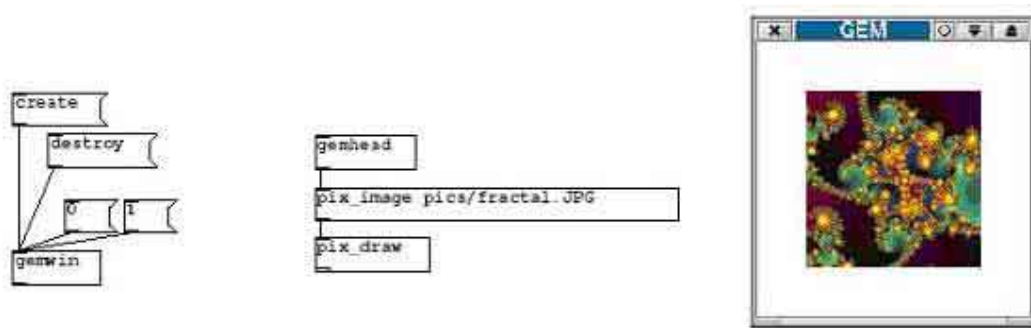


Figura 1: Interface de GEM, en donde se pueden ver los objetos representados por las cajas de textos y las líneas que definen las conexiones, el sentido de transmisión de la información es de arriba hacia abajo.

En general este tipo de software permite controlar sonido y/o video en tiempo-real a partir de parámetros que pueden obtener de sensores, instrumentos musicales, sistemas ópticos de captación movimiento, o una red informática. Uno de los más destacados es EyesWeb una plataforma desarrollada por el Laboratorio de Informática Musical de la Universidad de Génova (<http://www.eyesweb.org/>). Se especializa en la captación de diferentes patrones de movimiento y gestualidad del cuerpo humano y la interpretación musical, y esta orientado a la producción de sistemas multimedia interactivos para el análisis de movimientos escénicos en tiempo real y controlar la síntesis de sonido y la ejecución en vivo de instrumentos.

Como ejemplo de su aplicación podemos mencionar nuestra instalación interactiva “Espejo Espectral” (<http://www.proyecto-biopus.com.ar/instalaciones.html#Espejo>) en la que el usuario puede controlar en tiempo real al generación de imágenes a partir de su propia silueta.

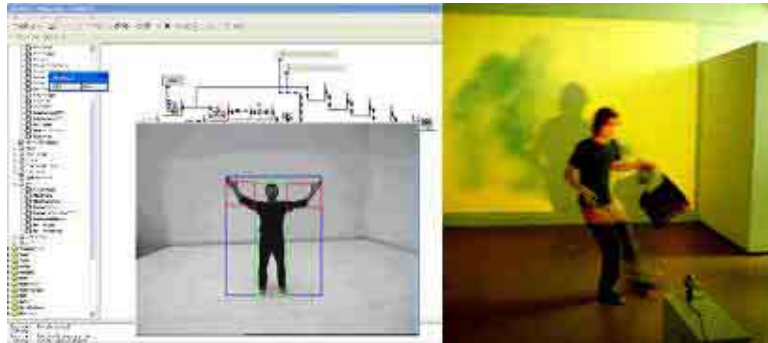


Figura 2: A la izquierda la interface de Eyesweb (imagen obtenida en <http://www.infomus.dist.unige.it/eywindex.html>). A la derecha, fotografía de la instalación Espejo Espectral del grupo Proyecto Biopus (www.proyecto-biopus.com.ar).

Otro software de esta línea es Pure Data (denominado también PD) <http://pd.iem.at/>, un entorno gráfico de programación creado por Miller Puckette, que corre en varias plataformas (Win32, OSX, IRIX y LINUX) y que en cierto modo es una versión libre de MAX. Tanto PD como EyesWeb, permiten extender su contenido a partir de la creación de nuevos objetos y funciones denominados “externals” o colecciones o grupos de objetos llamados librerías. Una de ellas, GEM (Graphics Environment for Multimedia) (<http://gem.iem.at/>), desarrollada por Mark Danks, es una librería que añade a PD la posibilidad de generar y manejar gráficos basados en Open-GL.

Sus puntos más fuertes son la generación, ejecución y procesamiento de audio digital. Pero la librería GEM posee objetos que le permiten vincularse a una cámara mini DV o webcam para realizar captura de movimiento de las personas en un espacio determinado.

3. Los lenguajes de programación para el arte interactivo

A diferencia de las aplicaciones anteriormente descritas, el conjunto de programas y librerías que abordaremos a continuación, están basadas en una programación por código escrito y no una configuración visual. Estos ambientes requieren un estricto conocimiento de la sintaxis (reglas de escritura), la que generalmente respeta la impuesta por el lenguaje C++. Este tipo de programación permite un mayor control y manejo de los recursos, especialmente si hablamos del procesamiento de la imagen píxel por píxel. Es este tipo de procedimientos lo que hace a estas herramientas tan potentes y lo que nos permite analizar una imagen, ya sea estática o de video, parte por parte, desde su mínima unidad, esto es píxel por píxel y desglosar su contenido en una serie de datos contenidos en una matriz para luego poder analizarlos y procesarlos con diversos algoritmos. Estos los hace particularmente útiles para el desarrollo de algoritmos de complejidad estructural en el

tiempo o el espacio, como la simulación de sistemas complejos y caóticos, la generación de fractales, reconocimiento de patrones o desarrollos de vida artificial.

En este grupo de software podemos nombrar el lenguaje de programación C++, particularmente las librerías para C++ hecha con fines artísticos: OpenCV, reactiVision y ArToolKit; y el lenguaje de programación (basado en Java) llamado Processing.

3.1. El lenguaje multiplataforma Processing

Processing (<http://www.processing.org/>) es un proyecto desarrollado por Ben Fry (Broad Institute) y Casey Reas (UCLA Design | Media Arts), un entorno y lenguaje de programación de código abierto (basado en Java) orientado a la programación de imágenes, animaciones y sonido en tiempo real, y está diseñado especialmente para realizar aplicaciones multimedia en entornos web e instalaciones. Es un lenguaje de gran potencia y muy bien documentado que se está extendiendo a través de diferentes plataformas como Windows, Linux y Mac, pero también existe una versión para telefonía celular y otra para programar unos microcontroladores (y placa digitalizadoras) llamados Wiring y Arduino. Processing está adquiriendo una gran prevalencia en la producción artística profesional y en la educación universitaria de arte electrónico a nivel mundial.

Uno de nuestros desarrollos con este lenguaje es una instalación llamada Pitch Fractal (<http://www.proyecto-biopus.com.ar/instalaciones.html#pitch>), en donde las personas pueden generar trazos a partir del canto, que a su vez generan figuras fractales. La obra combina algoritmos desarrollados en Max/MSP y Processing.

Es interesante mencionar como este ejemplo nos muestra la versatilidad de estas herramientas en relación a la captación del gesto, ya que en este caso no hablamos de gestualidad visual sino sonora.



Figura 3: A la izquierda captura de interface de Processing. A la derecha, fotografía de la instalación Pitch Fractal del grupo Proyecto Biopus (www.proyecto-biopus.com.ar).

Processing puede utilizar un plugin de código abierto llamado Myron que son una serie de librerías compiladas para diferentes plataformas y lenguajes. La versión para Java y Processing se conoce como JMyron y la de Macromedia Director, como WebCamXtra. El objetivo de este proyecto es mantener libre de costos las herramientas y técnicas de la visión artificial para su fácil utilización dentro de la comunidad artística y en la educación. Las tareas que realiza esta aplicación se concentran en la captación de movimiento, el reconocimiento y segmentación de la imagen.

Una de nuestras obras que implementa JMyron en processing es “Tango Virus” (<http://www.proyecto-biopus.com.ar/instalaciones.html#tango>), una instalación interactiva que permite al público modificar en tiempo-real un tema de tango.

El público puede bailar el tema de tango que se está escuchando, pero dicho baile se transforma en un comportamiento viral que ataca al tema musical, haciendo que este varíe, quizás al punto de “fallecer”.



Figura 4: A la izquierda esquema del dispositivo de captura de la instalación Tango Virus del grupo Proyecto Biopus (www.proyecto-biopus.com.ar). A la derecha, fotografía de la instalación.

3.2. El lenguaje C++ y la visión artificial

Históricamente el lenguaje C++ ha sido la herramienta de desarrollo de software por excelencia. Cuando se habla de C++, se habla de velocidad y potencia, no es de extrañar que en este sector en donde el ancho de banda es tan importante (debido al peso que implica el procesamiento en tiempo-real de sonido y video), sea este lenguaje un de sus opciones. En contraposición con su potencia, C++ requiere un gran conocimiento de programación.

A la hora de desarrollar este tipo de trabajos, el lenguaje C++ requiere de librerías especializadas (código pre-escrito para facilitar funciones). Estas librerías de visión artificial (CV - computer vision) trabajan con métodos que permiten a la computadora “comprender” las imágenes. Para un algoritmo corriente la imagen no es más que una colección de píxeles que no posee interpretación alguna,

desde esa perspectiva un rostro y un automóvil sólo difieren por los colores que contienen, pero no se deduce forma ni se obtiene ningún tipo de significado de ello. Los algoritmos de visión artificial apuntan a que la computadora pueda interpretar diferentes tipos de información, como por ejemplo la interpretación del espacio observado (distancias, obstáculos, perspectiva, etc.) o el reconocimiento de patrones (formas geométricas, rostros, etc.).

3.2.1. Las librerías OpenCV (Open Source Computer Vision)

<http://www.intel.com/technology/computing/opencv/index.htm>

Originalmente desarrolladas por la empresa Intel, se encuentra dentro del primer grupo, y permite realizar diversas tareas tales como la identificación de objetos, segmentación y reconocimiento de una imagen, reconocimiento facial, seguimiento de trayectorias y movimiento, captación del gesto.

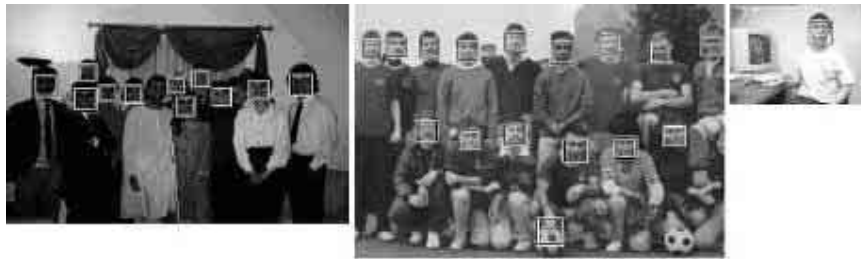


Figura 5: reconocimiento de rostros (imagen obtenida en "Rapid Object Detection using a Boosted Cascade of Simple Features" de Paul Viola and Michael J. Jones)

3.2.2. La Realidad Aumentada y las librerías ARToolKit (Augmented Reality Tool Kit)

<http://www.hitl.washington.edu/artoolkit/>

ARToolKit es una librería para la construcción de aplicaciones de Realidad Aumentada (Augmented Reality –AR-), aplicaciones que implican la superposición de imágenes virtuales con las del mundo físico circundante. Una de las dificultades en el desarrollo de este tipo de aplicaciones es el seguimiento del punto de vista del usuario, para saber desde que perspectiva se construirá la imagen virtual, la aplicación necesita saber como el usuario esta mirando en el mundo exterior. ARToolKit utiliza algoritmos de visión artificial (CV) para resolver este problema, sus librerías de seguimiento de video, calculan en tiempo real la posición y orientación de la cámara, usando patrones bitonales impresos sobre los objetos de los que necesita conocer su ubicación y perspectiva.

Alguno de los proyectos que podemos mencionar es The Mixed reality Lab (<http://www.mixedrealitylab.org/>), el cual no solo se especializa en el desarrollo de imágenes virtuales en 3D como vemos en la siguiente figura, sino que incorpora un novedoso método para la generación de imágenes virtuales 3D a partir de señal de video.



Figura 6: Realidad Aumentada (imágenes tomadas de MXR Research Lab)

La incorporación de contenido 3D en vivo dentro de aplicaciones AR se basa en un algoritmo de reconocimiento de la forma a partir de la silueta (shape-from-silhouette algorithm). El objeto o sujeto a capturar es monitoreado por 15 cámaras, el sistema genera una vista original, que puede ser generada a una velocidad interactiva, y que se incorpora (o se incluye) dentro del espacio de AR. El objetivo principal de este proyecto es producir un modelo humano tridimensional real que pueda ser utilizado en entornos virtuales.

3.2.3. ReactiVision y el reconocimiento de patrones

ReactiVision son unas librerías para el reconocimiento y seguimiento de patrones bitonales prediseñados. Se diseñó principalmente como una herramienta para la creación de mesas tangibles a modo de interfaces. Basada en otras librerías, esta aplicación fue desarrollada por Martin Kaltenbrunner, en el Music Technology Group, en Barcelona, como parte del proyecto de reactTable, un instrumento musical que emplea las mesas tangibles anteriormente mencionadas.

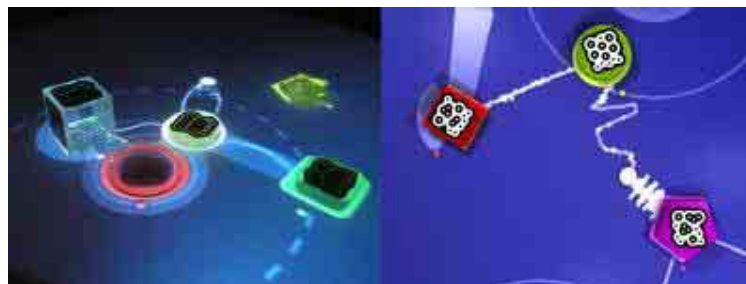


Figura 7: ReactiVision, en la imagen de la derecha pueden observarse los patrones bitonales.

Estas mesas tangibles están constituidas por una superficie translúcida, sobre la cual se colocan diversos objetos que tienen la impronta de las diferentes marcas a reconocer. ReacTiVision registra la posición de las marcas (usando una cámara) y envía los datos vía OSC (OpenSound Control), un protocolo de comunicación que puede leerse con Processing, C++, EyesWeb y otros. Luego unos de estos software construye la respuesta visual y sonora del sistema en función de esta información.

4. Conclusión

La explosión, dada durante los últimos años, de herramientas de software open source destinadas al desarrollo de nuevas interfaces, promete un futuro cercano de nuevas formas expresivas, que resignificarán el lugar del cuerpo y la experiencia del espacio y el tiempo en el arte.

La evolución del software open source, insisto, no puede entenderse aparte de un desarrollo equivalente en el hardware, en donde se recurre a la re-ingeniería para dar nuevos usos y utilidades a dispositivos obsoletos, o se modifican otros de bajo costos, como es la transformación de cámaras web comunes en cámaras infrarrojas, o el desarrollo de pantallas sensibles al tacto de bajo costo, gran tamaño y con multitacto, utilizando cámaras, leds y acrílico.

De todos modos, creemos que el avance más importante que impone este movimiento es el de la transmisión del conocimiento, la sinergia de miles de diseñadores, productores, experimentadores que reutilizan y reinventan la tecnología a pasos vertiginosos. En este sentido, las tecnologías open source han generado un clima de “nuevo renacimiento” coherente con el sincretismo actual entre arte, ciencia y tecnología; un momento que, sin duda, tendrá una importancia histórica en el arte, equivalente a las que han tenido las vanguardias del siglo XX.

Esta obra está licenciada bajo una Licencia Atribución-NoComercial-SinDerivadas 2.5 Argentina de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/2.5/ar/> o envíenos una carta a Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.