



Documents of 20th-century Latin American and Latino Art

A DIGITAL ARCHIVE AND PUBLICATIONS PROJECT AT THE MUSEUM OF FINE ARTS, HOUSTON

ICAA Record ID: 1093980

Access Date: 2017-04-10

Bibliographic Citation:

Hernando Barragán. "Software: ¿arte?" In *Hipercubo/ok/: arte, ciencia y tecnología en contextos próximos*, 188- 195. Bogotá, Colombia: Comité de Investigaciones de la Facultad de Artes y Humanidades, Facultad de Ingeniería, Universidad de los Andes, 2002.

WARNING: This document is protected by copyright. All rights reserved. Reproduction or downloading for personal use or inclusion of any portion of this document in another work intended for commercial purpose will require permission from the copyright owner(s).

ADVERTENCIA: Este documento está protegido bajo la ley de derechos de autor. Se reservan todos los derechos. Su reproducción o descarga para uso personal o la inclusión de cualquier parte de este documento en otra obra con propósitos comerciales requerirá permiso de quien(es) detenta(n) dichos derechos.

Please note that the layout of certain documents on this website may have been modified for readability purposes. In such cases, please refer to the first page of the document for its original design.

Por favor, tenga en cuenta que el diseño de ciertos documentos en este sitio web pueden haber sido modificados para mejorar su legibilidad. En estos casos, consulte la primera página del documento para ver el diseño original.

Synopsis:

This text, "Software: ¿Arte?" [Software: Art?] is part of the book *Hipercubo/ok/: arte, ciencia y tecnología en contextos próximos* [Hypercube/ok/: Art, Science, and Technology in Neighboring Contexts] that was produced by Andrés Burbano and Hernando Barragán in 2002. The book is a collection of essays written by several authors about contemporary art in Colombia. This one discusses the use of software and computer code as a form of artistic expression, keeping in mind its artificial nature. Based on its recent use by Colombian artists and its connection to net art and computer art, Barragán is curious about the relationship potential of computer code (a preexisting language used by the programmer to communicate with the machine) and algorithms (procedures that lead to a result) at two distinct levels, whether distributed on the Internet (World Wide Web) or embedded in software programs. He sees the former as an "interactive environment in real time," and the latter as a "high-level alphabet" that can take on new meanings as a variety of discrete elements connect or overlap. Barragán discusses the role of this artificial language, the medium in which it is used, and its digital expression via an interface in terms of the aesthetic possibilities of the work as viewed within a contemporary creative context.

hipercubo/ok/

< arte, ciencia y tecnología en contextos próximos >



< Andrés Burbano, Hernando Barragán >

processing: 20%

Escriben en HiperCubo/ok/
Rodrigo Alonso, Hernando Barragán, José Luis Brea, Andrés Burbano, Alejandro Duque, Gabriela Häbich,
Eduardo Kac, Rafael Lozano-Hemmer, Brian MacKern, Sergio Pineda, Tania Ruiz, Andrés Sicard,
Francisco Varela.



hipercubo/ok/

arte, ciencia y tecnología en contextos próximos

Andrés Burbano
Hernando Barragán

Comité de Investigaciones de la Facultad de Artes y Humanidades
Facultad de Ingeniería
UNIVERSIDAD DE LOS ANDES

GOETHE INSTITUT BOGOTÁ

S



:

T T S S
F S

F T
S

O

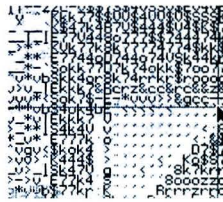
F

S

F

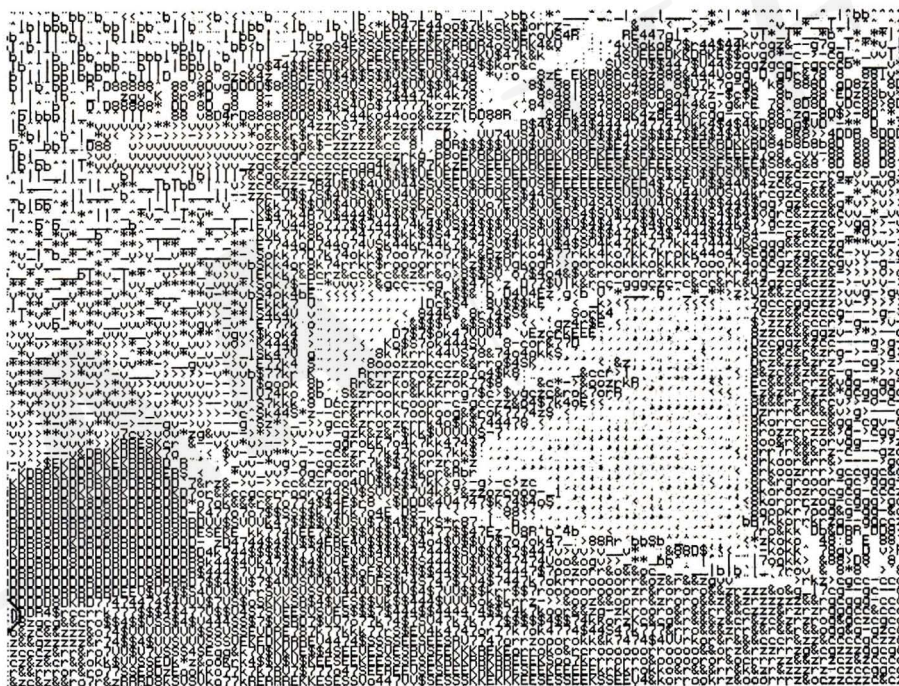
O F
T F

.SOFT



Software: ¿arte?

<Hernando Barragán>



187

.soft

Para comenzar a hablar de *software* debemos hablar de “código”, y para esto debemos primero definir algunos términos que son relevantes.

Algoritmo: es un procedimiento, o conjunto de reglas, expresado usualmente en notación matemática. Es un conjunto sistemático de pasos, procesos u operaciones que producen un resultado específico como la respuesta a una pregunta o la solución de una ecuación.

Código: es el lenguaje preestablecido entre el programador y la máquina, mediante el cual ponemos en términos inteligibles por ésta, nuestras manifestaciones de creatividad; es pertinente aclarar que los lenguajes de programación y el código escrito en estos lenguajes (instrucciones codificadas) no son para comunicarse, sino para dar instrucciones a la máquina. La diferencia está en que no se pretende enviar un mensaje a la máquina, sino darle unas órdenes, y en que la máquina no puede usar este lenguaje para respondernos. La posterior interpretación y ejecución del código por parte de la computadora, materializa la posibilidad de una experiencia para el espectador (usuario) siguiendo un algoritmo, que paso a paso, nos permite estar más cerca de alcanzar nuestro objetivo en particular.

Este código usualmente nace de la implementación de un modelo, proceso mediante el cual un modelo es colocado en términos de estructuras e instrucciones en un lenguaje inteligible por la máquina. Cabe destacar que este modelo no necesariamente proviene de la realidad, pues la computación nos permite a diferencia de otros medios, construir posibilidades en las que las reglas que aplican a nuestro universo se pueden evadir. El código es usualmente organizado en fragmentos modulares con tareas específicas, en los que cada una de sus líneas escritas tiene un propósito, es acercarnos a una “solución” de un problema en el mundo de los *bits*, independiente de su posibilidad en el mundo de los átomos.

Sacar el código del contexto usual del *software* y verlo en un contexto artístico, provoca una forma diferente de pensar acerca del *web*, el *software* y el concepto mismo de la red. Todas estas cosas son ahora un fenómeno cultural. La curiosidad por revelar los mecanismos por los cuales el computador produce sus ilusiones como parte de una obra de arte o de una nueva tecnología, trajo consigo lo inevitable: revelar el código, revelar las estructuras que hacen posible la magia. Así que, ¿cómo interactuar con todas estas cosas?

Empezaré con un fragmento traducido de un artículo de Florian Cramer and Ulrike Gabriel (2001).

188

.soft

Desde hace más de una década existen festivales, premios, exhibiciones y publicaciones para varias formas de “computer art”, música electrónica, computación gráfica, literatura electrónica, “net art”, e instalaciones interactivas controladas por computador, para nombrar sólo algunas, cada una con sus instituciones y discurso. Clasificaciones como las anteriores muestran claramente que la atención es usualmente prestada al cómo (en cuál medio) las obras de arte son presentadas a la audiencia externamente. Éstas también muestran cómo el arte digital es tradicionalmente considerado como parte del “[new] media art”, un término que cubre medios análogos y digitales, históricamente ligado al video arte. Pero, ¿no se está asumiendo erróneamente que el arte digital, (el arte que consiste de ceros y unos) se derivó del video arte, sólo porque los datos en la computadora son convencionalmente visualizados en pantallas?

Al llamar arte digital “[new] media art” la percepción del público, particularmente se ha enfocado en los ceros y unos en forma de medios visuales, acústicos y táctiles, en vez de estructuras de programación. Esta percepción es reforzada por el hecho de que los algoritmos empleados para generar y manipular la música en el computador, la computación gráfica, el texto digital, son frecuentemente, si no en la mayoría de los casos invisibles, desconocidos para la audiencia y los artistas. Mientras que la historia del arte computacional es corta, ésta es rica en trabajos cuya programación reside en cajas negras o es considerada simplemente como un proceso preparatorio detrás de escenas para un trabajo terminado (y finito) en un CD, en un libro, en Internet o en un ambiente interactivo en tiempo real. Mientras que el código está inevitablemente trabajando en todo el arte que es digitalmente producido y reproducido, éste tiene una larga historia de ser pasado por alto como material artístico y como un factor más en el concepto y la estética de un trabajo. Esta historia va de la mano con la evolución de la computación desde sistemas que podían ser usados solo por programadores, a sistemas como el Mac o el Windows, los cuales por su interfase gráfica de usuario, camuflan el hecho de que ellos están ejecutando un código de programación, tanto en su operación, como en su estética, y esto nos lleva a reflexionar sobre cómo la realidad es moldeada por el *software*.

Debido a que el *software* es código de control de máquina, se deduce que los medios digitales son literalmente, escritos. La literatura electrónica no es simplemente texto, o híbridos de texto y otros medios, circulando en las redes de computadores. Si la literatura es algo hecho por letras, el código de programación, protocolos de *software*, y formatos de archivos de las redes de computadores, constituyen una literatura cuyo alfabeto detrás es ceros y unos. Ejecutando el código en sí mismo, este código es constantemente transformado en alfabetos de alto nivel, legibles por humanos, o letras alfanuméricas, píxeles gráficos y otros significantes. Estos significantes fluyen una y otra vez, de un formato a otro. Los programas de computador están escritos en una sintaxis altamente elaborada, de múltiples capas de código mutuamente interdependientes. Esta escritura no solamente está presente en los sistemas de computación como medio de transporte, sino que activamente los manipula cuando es instrucciones de máquina. La diferencia es obvia cuando se compara un mensaje de e-mail convencional con un virus e-mail: aunque ambos son pequeños fragmentos de texto cuyos alfabetos son los mismos, el virus contiene código con sintaxis de control de máquina, que interfiere con el código del sistema al cual es enviado. “*Software art*” significa un cambio en el artista desde su visión de pantallas a la creación de sistemas y procesos en sí mismos; esto no está cubierto por el concepto de “media”. “Multimedia”, como un término paraguas para formatear y mostrar datos, no implica por definición que los datos son digitales y que el formateo es algorítmico” [1].

Con respecto a este artículo, en charlas con S. Brooks, vemos cómo es cierto lo que se dice en el artículo respecto a la errónea clasificación de los medios digitales, pero encontramos también algunos aspectos que son discutibles:

Lo que se dice en este artículo acerca de lo que se llama “media art” es correcto en cuanto a “media art” ha sido definido en términos de lo visual y áureo más que en el código que lo soporta. No estoy seguro si esto es porque históricamente está ligado al video arte, como escriben los autores. Más bien hay una relación (histórica y formal) con el cine (de la cual supongo también se desprende el video arte), pero no diría que esta relación cinematográfica es la única o la única raíz, eso sería simplista e injusto. Hay también una relación con los juegos y la metáfora del movimiento físico. También es injusto, que para ellos introducir sus intereses en “*software art*” lo hagan como una clase de escritura, estoy en desacuerdo con su literalismo cuando afirman: “Debido a que el *software* es código de control de máquina, se deduce que los medios digitales son literalmente, escritos” [2].

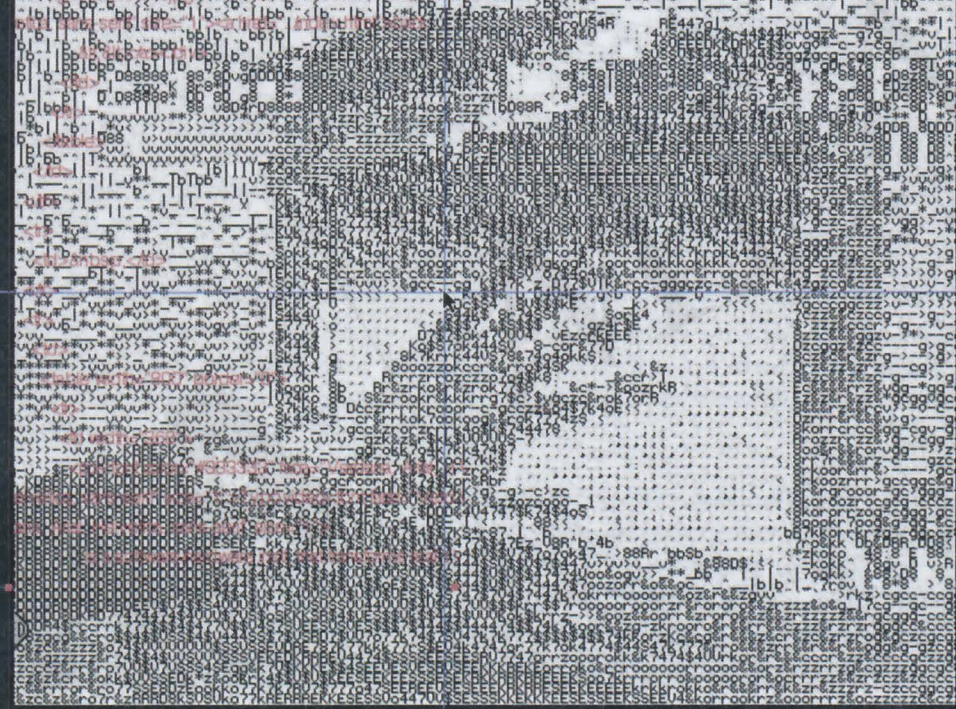
En un ensayo enviado por S. Brooks y B. Stryker al CAA (College Art Association Annual Conference) en febrero 22 de 2002 [3], ellos señalan que escribir datos, o sentarse a escribir en un computador son apropiaciones erróneas, en otras palabras, no es realmente “escribir” lo que nosotros hacemos y no es realmente “escribir” lo que las computadoras hacen con los datos, en vez de eso deberíamos hablar más bien de mover y copiar datos.

En el mundo de los *bits*, o mundo digital ocurren muchos procesos en tamaños y velocidades cuyas escalas están bastante lejos de nuestro sentido de la experiencia. En el transcurso de nuestra interacción con las representaciones digitales, la mayoría de nosotros no tenemos percepción de los *bits*, estructuras de código, algoritmos, o siquiera de recordar que existen. Es aquí donde podemos ver que la experiencia de la representación digital necesita ser aún más explorada. Pero estas operaciones, aunque son imaginables, son en sí mismas inalcanzables a nuestros sentidos, y más aún en éstas hay muy poca semejanza cuando las comparamos con la forma como nosotros interactuamos con el mundo físico, sólo al final somos capaces de percibir sus efectos. En el mismo instante en que estas operaciones trabajan de manera invisible detrás de escena, realizando estas transformaciones, parece un acto de magia, y es así como el medio a través del cual se posibilita el cambio no tiene forma en si mismo y, es entonces, operación pura. Provee una función sin pérdida ni ganancia en el proceso. En este sentido los medios digitales se perciben inmateriales, parecen pura función, intocables por la forma. El código es entonces un lenguaje con posibilidades estéticas y de forma, una codificación que tiene un poder transformador que en el medio digital opera transformaciones, que a su vez podrían generar nuevas formas.

Una operación que el almacenamiento digital y la computación hacen posible es la de ser un transformador universal de información desde una manifestación física en otras de naturaleza totalmente distinta. Es así como el sonido una vez grabado digitalmente puede ser convertido en luz sobre una pantalla, en el movimiento de

Typovideo, una experiencia video en ASCII, visto desde el Hiperlook
<http://sie.uniandes.edu.co/~h/typovideo.html>

URL: <http://sie/~h/typovideo/index.html>



URL: <http://sie/~h/typovideo/index.html>

unos objetos físicos, o en un sinnúmero de fenómenos sensibles. Con la computación digital, entendida ésta como el uso y programación de computadores digitales; la especificidad y esencia de una forma de experiencia, por ejemplo la voz, se disuelve cuando es transformada en otra, como imágenes, y esto sugiere que puede haber formas de conectar cosas muy diferentes. Una mezcla de espacios de manifestación artística.

Otro aspecto muy importante de la computación digital es la *interactividad*, que se ha convertido en una de las expresiones más importantes de la civilización contemporánea. En computación ésta involucra el diálogo entre la máquina y el usuario. La adopción de la computación como un medio del arte ha generado nuevas maneras interactivas de forma y contenido, lo cual ha permitido que a su vez la computación misma pueda ser vista con otros ojos, y así productos de la computación sean reevaluados en otro contexto fuera del *software* como herramienta, específicamente en el contexto del arte. Para un programador, el proceso artístico es el acto mismo de escribir el código.

Aún no hay reglas que afecten el futuro del código como trabajo de arte, así que hay todo un universo de posibilidades por delante. ¿Desde dónde tomar una posición y hacer una valoración de un trabajo de arte de este tipo?, tan solo desde el punto de vista de la interactividad que ha contribuido a brindar una experiencia a nivel global, podemos citar ejemplos concretos. Es así como en 1999 vimos ganar en la categoría .net del prestigioso concurso de arte electrónico Ars Electronica al sistema operativo linux, poniendo intencionalmente sobre la mesa la discusión de si el código en sí mismo puede ser una obra de arte.

“La categoría .net no es un premio para el sitio *web* más hermoso o el más interesante de la red... la intención es también iniciar una discusión y analizar en qué punto el código mismo puede ser un trabajo de arte” (A. Broeckmann). Este autor, historiador cultural, quien fuera jurado el año anterior a la premiación en la categoría .net menciona: “El premio Ars no es tanto acerca del arte como es acerca de la fusión creativa de tecnología y creatividad humana, sin duda hay una dimensión política en otorgar el premio a un sistema operativo que es no comercial. Pero el premio también reconoce algunos de los aspectos que caracterizan una buena pieza de arte en la red, y es la de ser realmente interactiva” [4].

Otro ejemplo que tomó por sorpresa a los medios fue la presentación de un virus en acción como obra de arte en la bienal de Venecia en junio de 2001. Un computador infectado con “biennale.py” permaneció en exhibición hasta que ésta cerró en noviembre. Las personas podían ver en tiempo real cómo los archivos y el sistema operativo de un computador eran consumidos y destruidos por el virus.

Esta pieza fue creada por la colectividad europea de “net art” 0100101110101101.ORG [5], en colaboración con epidemiC, otro grupo conocido por sus habilidades en programación. El virus solo afecta programas escritos en el

lenguaje de programación Python y se esparce si alguien baja *software* infectado o usa un disquete corrupto. La difusión de este virus en la red generó una gran polémica a todo nivel, aunque sus creadores advirtieron a compañías acerca del virus y de cómo quitarlo de sus sistemas, los expertos en seguridad en la red no están de acuerdo con el uso de estos métodos para ganar atención. Otro caso que generó reacciones de todo tipo fue el uso del “*spam*” como forma de arte, y es así como la colectividad artística Jodi.org envió 1039 mensajes de *spam* en la lista de correo Rhizome Raw en enero de 2001.

En 2001 aparece el premio “*software art*” del Transmediale Art Festival, que no solo es el primero de su clase en este festival de arte en particular, sino el primero de su clase en general. Apartes de la declaración del jurado:

Hasta donde sabemos, este es el primer premio dado y dedicado exclusivamente al *software art*. Este premio no es acerca de lo que comúnmente se entiende como multimedia donde el enfoque está sobre datos que pueden ser visualizados, escuchados o sentidos. Este premio se trata de los algoritmos, es acerca del código, que genera, procesa y combina lo que podemos ver, oír, y sentir. Mientras que el código de una imagen, sonido, o texto se apoya en otras piezas de *software* para ser perceptible y editable, el código de programación activamente manipula la máquina. Tal vez el aspecto más fascinante de la computación es que el código, sea mostrado como texto, o como números binarios, puede ser ejecutable por la máquina, un fragmento inocuo de escritura puede disgustar, reprogramar, acabar el sistema (...).

Los virus de computador pueden ser vistos como una forma crítica de “*software art*”, porque ellos hacen que los denominados usuarios noten que el código digital es virulento. El *software* de computadoras no es simplemente una herramienta. Cada programa que pretende ser una herramienta... Usted espera que “*Save*” guarde su archivo y no lo borre. La sensación de entender y controlar lo que el *software* hace en la máquina puede basarse solamente en la confianza en el programador.

Para nosotros *software art* se opone a la noción de *software* como una herramienta (...) El *software art* tiene el potencial de hacernos dar cuenta de que el código digital no es inofensivo, que no está restringido a simulaciones de otras herramientas, y que es en sí mismo, una base en la práctica creativa” [6].

En todos estos casos, productos de la red, para y por la red, fabricados de código, el material real de la red, una simbiosis exitosa de la tecnología y la creatividad humana. Como un nuevo medio, la computación ofrece un potencial enorme, pero su exploración va a un ritmo lento debido a la falta de entendimiento de la tecnología por parte de los artistas y diseñadores, y también debido a la

desconexión del arte y el diseño por parte de los ingenieros y tecnólogos. Existe un amplio espectro de productos de *software* que han sido diseñados y usados “imitando” en el computador el desarrollo de actividades tal y como las realizamos en el mundo físico (pintar, dibujar, escribir), tanto en su manipulación como en su interacción y desarrollo, sin explorar la posibilidad del medio digital. Sin embargo, vemos cómo cada día se abre paso una nueva generación, híbrida, que está permitiendo esta conexión y con ello el surgimiento de nuevas escalas de medida y criterios que permitirán reevaluar el estado y la práctica de muchas disciplinas y del quehacer artístico.

Hablar del contexto “artístico” del código suena algo extraño, pero debemos comprometernos en la búsqueda y experimentación de buenos trabajos que reflejen el pensamiento creativo contemporáneo. Para ello debemos poder usar todas las formas de expresión de la cultura contemporánea, y si nos preguntamos cuál es la forma más pura del pensamiento creativo contemporáneo, pienso que es el código, *software* es arte.

Notas

- [1]. Florian Cramer and Ulrike Gabriel. *Software Art*. August 15, 2001. Mi traducción.
- [2]. Comunicación personal, 2002.
- [3]. S. Brooks y B. Stryker. “Spaces of Exile, Technically Speaking”. Paper submitted to CAA (College Art Association Annual Conference) February 22, 2002.
- [4]. Andreas Broeckmann, Jurado del concurso *Ars Electrónica* categoría .net 1999.
- [5]. <http://WWW.0100101110101101.ORG> biennale.py
- [6]. <http://www.transmediale.de/01/en/software.htm> Jury Statement 2001.