

Festival Transmediale.01 (Berlín, 8 de febrero de 2001)

¿El software artístico es material artístico genuino?

Andreas Broeckmann

abroeck@transmediale.de

Director Artístico de transmediale

Florian Cramer

cantsin@zedat.fu-berlin.de

Profesor de Literatura Comparada en la Universidad Libre de Berlín

Ulrike Gabriel

gabriel@em.uni-frankfurt.de

Directora de Codelab Berlin

Golan Levin

golan@flong.com

Profesor de Diseño Interactivo

Rafael Lozano-Hemmer

rafael@lozano-hemmer.com

Artista digital

Anne Nigten

anne@v2.nl

Gestora artística

Daniela Plewe

plewe@is.in-berlin.de

Artista digital

Gerfried Stocker

gerfried.stocker@aec.at

Codirector gerente del Ars Electronica Center

Resumen: En el marco del festival *transmediale.01* se organizó el primer concurso que incluye un premio de arte a la programación informática. Este concurso reconoce el trabajo artístico que hacen los artistas programadores que no son ni *artistas multimedia* ni *artistas en red*, cuyo material estético es el código y cuya forma de expresión es programar. Una posible definición de *software artístico* es la que incluye proyectos en los que el software algorítmico escrito por los propios artistas (programas autónomos o aplicaciones basadas en secuencias de órdenes) no es simplemente un instrumento funcional sino una creación artística por sí mismo. ¿Tiene el software una función puramente instrumental o bien ofrece nuevas perspectivas culturales y creativas? ¿El código informático es material artístico genuino como lo son las imágenes digitales o las imágenes pictóricas?

Andreas Broeckmann: En el marco del festival *transmediale.01* se ha organizado el primer concurso que incluye un premio de arte a la programación informática. Este concurso reconoce el trabajo artístico que hacen los artistas programadores que no son ni *artistas multimedia* ni *artistas en red*, cuyo material estético es el código y cuya forma de expresión es programar. Una posible definición de *software artístico* es la que incluye proyectos en los que el software algorítmico escrito por los propios artistas (programas autónomos o aplicaciones basadas en secuencias de órdenes) no es simplemente un instrumento funcional sino una creación artística por sí mismo. ¿Tiene el software una función puramente instrumental o bien ofrece nuevas perspectivas culturales y creativas? ¿El código informático es material artístico genuino como lo son las imágenes digitales o las imágenes pictóricas?

Gerfried Stocker: Para mí es interesante observar que, incluso en este estadio avanzado de tratar con el software artístico, siempre aparece la extraña necesidad de justificar ese tipo de arte fuera del contexto de arte tradicional. Y creo que no es suficientemente radical... no tan radical como usted quería...

Andreas Broeckmann: Creo que subestima el desarrollo del arte durante los últimos treinta años. Me parece que lo que usted califica de nuevo y radical en el software artístico es el tipo de radicalidad que reivindican todas las vanguardias. Estoy seguro de que en estos momentos habrá tres o cuatro vanguardias en marcha que reivindican la misma radicalidad. Y sigue siendo arte.

Gerfried Stocker: Por supuesto que se trata de arte, pero muchas cosas son arte. Todo el mundo sabe que interpretar a Mozart en la Staatsoper de Viena es un arte maravilloso. De eso no cabe la menor duda. Sabemos que ver películas en la Berlinale es arte, y lo llamamos así. El arte es un concepto demasiado amplio para definirlo y, de hecho, no nos interesa suficientemente para ese tipo de desarrollo. Debemos tener suficiente confianza para ver que el software artístico es una reacción contra el desarrollo general de la sociedad y de la cultura dirigido por las nuevas tecnologías, la digitalización, la conversión de cualquier cosa en un código. Los artistas tienen muy pocas oportunidades de trabajar en él si quieren mantenerse dentro del proceso y de su transformación dinámica. En caso contrario, salen de él y realizan obras artísticas interesantes, como el caso de Jeffrey Shaw o Bill Viola. Son unos artistas fantásticos, pero son ejemplos de lo que estoy diciendo...

Ulrike Gabriel: Pero ahora no vamos a hablar de ellos porque no programan....

Gerfried Stocker: Seguro que aquí todo el mundo conoce la obra que Bill Viola realizó con motivo de la inauguración del ZKM. Tienes una pantalla delante y cuando te diriges hacia ella, hay un árbol que crece, envejece y muere según la distancia a la que uno se encuentra de la pantalla. Hace unos meses me encontraba en un congreso en Stanford, y Bill Viola explicó la obra y reivindicaba que se trataba de arte interactivo. Y hablaba de sí mismo como artista y de Bernd Lintermann –un programador informático del ZKM que trabajó con él. En mi opinión, en este proyecto el arte auténtico lo realiza Bernd Lintermann, porque es quien escribió el código que hace que el árbol crezca, se transforme y muera. Y lo que hizo Bill Viola fue dejar a un lado su papel de artista y limitarse a ser una especie de escenógrafo. Porque lo único que hizo fue decidir la apariencia que la obra iba a tener dentro del museo.

Daniela Plewe: Estoy de acuerdo con usted y creo que hay varias formas de especificar el código. Me parece que estoy de acuerdo en que decir: "Aquí nos hace falta un árbol o allí nos hace falta algo que tome decisiones" no es suficiente. En mi caso, por ejemplo, siempre he estado interesada en aprender formalismos muy abstractos. Y tardé mucho más de lo que pensaba. Posiblemente también deberíamos hablar de esta delegación del código y de los inconvenientes que implica. Es evidente que se pierde la respuesta de calidad ante el material artístico. Eso no voy a negarlo, y lo cierto es que se trata de una experiencia dolorosa. Si te comunicas con el programador, es evidente que estas personas se implican mucho más en el concepto estético cuando todo está listo. Otra cuestión es si pretendemos la misma estructura que en la producción de una película, donde se acepta totalmente la distribución del trabajo, y cómo crear software artístico en el futuro.

Golan Levin: Mi experiencia anterior fue como pintor y compositor. No fue hasta después de haber sufrido numerosas malas experiencias a la hora de intentar convencer a los informáticos para que me ayudaran a construir mi obra que, frustrado, tuve que aprender a hacerlo yo mismo. Les pedía que me ayudaran en determinadas cosas y ellos me decían: "Cobro cien dólares por hora" o "Te ayudaré por lástima", o bien apoyaba mi cabeza en su hombro y decía: "Eso no funciona estéticamente", y ellos me decían: "No entiendo a qué te refieres." Al final no tuvimos otro remedio que ponernos manos a la obra. No sólo es un trabajo laborioso, aunque este trabajo también es importante. Hay que estar en contacto con él sin olvidar la parte conceptual.

Ulrike Gabriel: Me gustaría volver a algo que ha dicho Gerfried Stocker. Como artista, utilizas un determinado material. En el proyecto *Vexation 1* de Antoine Schmitt había una pantalla negra con una bola que iba rebotando; se habían utilizado unos determinados materiales, se había abierto un determinado espacio. El espacio era mínimo. Y dentro de este espacio se utilizaba el material artístico, con esta definición de parámetros. Eso estaba bien. En el ejemplo de Gerfried, Bill Viola ni siquiera sabía qué estaba utilizando. Realizaba un trabajo que no podía realizar como artista. Por este motivo, tomo una postura muy elemental: seamos artistas. Simplemente vayamos, toquemos el código, maneámoslo, utilicémoslo en nuestra obra de arte, considerémonos artistas en el sentido clásico del término, porque de esa forma mantendremos el proceso.

Gerfried Stocker: Estoy totalmente a favor de cualquier tipo de software artístico. Esto no es una feria de arte donde vendemos software artístico; somos una comunidad intelectual que habla de problemas y temas relacionados con el software artístico. Y debemos ser más radicales y exigentes. No debería ser suficiente que uno simplemente escribiera código. Eso no es suficiente para ser artista programador. Algunas obras son muy próximas al enfoque –aunque importante– de los artistas que hacen un uso superficial de la tecnología: "Tenemos el control de las herramientas y eso es importante." Realmente es importante, pero se trata de una dirección distinta. Me interesa mucho un tipo de obra que realmente utiliza el software de manera que refleje una nueva forma de producir y hacer arte, en el sentido de pasar del documento al evento. Cualquier imagen digital, cualquier sonido digital, cualquier muestra es una secuencia de órdenes, pero no es realmente software. Sigue siendo un documento. Y entonces llegamos al evento: una forma de descripción de algo que realmente es el algoritmo que se reproduce cada vez. Es un proceso que consiste en hacer que el texto codificado en una memoria de silicio llegue a ser algo.

Público - Tim Druckrey: Hay dos cosas que deberíamos dejar claras. Una es que el software no solo es el código, sino también el algoritmo. Si programo algo en C pero quiero colgarlo en la web y no sé programar en Java, y encargo que escriban el código en Java, ¿significa esto que he renunciado a mi papel de artista? Si formulo el algoritmo para que alguien lo codifique, ¿son ellos los artistas o soy yo el artista?

Daniela Plewe: He anotado lo que ha dicho más o menos literalmente. Me gustaría poner sobre la mesa la noción de arte algorítmico en lugar de software artístico.

Golan Levin: Yo utilizo algoritmos en mis obras y no niego que a veces el resultado de dichos algoritmos es magnífico, pero sospecho que si tuviera que participar en un concurso de algoritmos en estado puro sería bastante aburrido. Si no, vaya y vea la colección de cinco volúmenes de algoritmos informáticos de Donald Knuth. No son más que una herramienta que podemos utilizar. Y se podría organizar un concurso, pero creo que haría falta un público realmente especializado que apreciara que los algoritmos de clasificación *quick* son mejores que los algoritmos *retex* o *bubble*. ¿Y por qué deberíamos dar un premio de arte a esas personas?

Público - Tomiko Thiel: Las obras que hemos visto esta noche son principalmente el tipo de herramientas que generan experiencias sensoriales. Posiblemente la de Daniela Plewe es la que ha conseguido con más éxito hacer que los datos y las estructuras de decisión sean visibles, que es lo que el jurado dijo que buscaba o esperaba encontrar en este concurso.

Florian Cramer: Es posible que a veces la experiencia sensorial de las muestras de software artístico que acabamos de ver permita ver los procesos generadores que tienen lugar en el interior. Si observan las presentaciones de esta noche, quizá tengan la sensación de que se trata de arte multimedia tal como lo conocemos. Pero imaginen que hubiéramos incluido un algoritmo autorreplicador, por ejemplo. Les habríamos podido enseñar una muestra de código C++ o algo parecido, pero no habrían entendido nada. Los trabajos estaban en el umbral entre el arte multimedia convencional, que básicamente se centra en la representación visual y sensorial, y unas obras de software artístico relacionado con los conceptos, con los algoritmos. ¿Hay alguna posibilidad de traducir esto en una experiencia sensorial? Me encantaría que algún día pudiéramos prescindir de las sensaciones producidas por los *rayos de datos*. Los rayos de datos son un aspecto del arte multimedia que realmente odio desde que existen, y todo se reduce a las presentaciones en forma de rayos de datos. Creo que nos dirigimos hacia lo invisible. Es realmente emocionante. Nos dirigimos hacia procesos que ya no se centran en la representación de interfaces. Que actualmente dispongamos de muchos ejemplos de software artístico que parten de la experiencia de la interfaz, pero la subvierten o la invierten, marca una especie de umbral, el final de un discurso y posiblemente el principio de otro discurso en arte electrónico.

Gerfried Stocker: Seguiré con la cuestión de la estética en el software artístico porque se trata de una expresión muy problemática: está demasiado vinculada al arte tradicional, pero es muy interesante fijarnos en qué pensamos cuando hablamos de estética en la programación. Quisiera referirme a la obra de Golan Levin, que admiro muchísimo. La estética de su código no está en las preciosas imágenes o agradables sonidos que crea. Lo que ha conseguido es crear algo parecido a la arcilla. La interfaz se integra completamente: es parte del propio software, del código y de su expresión. Y creo que éste es uno de los mayores éxitos del arte informático, independientemente de lo que produzca, tanto si al final son unas imágenes bonitas, unos sonidos agradables o posiblemente algún virus tan bien creado que es capaz de acabar con Internet.

Andreas Broeckmann: Sin embargo deberá explicar por qué esta plena integración como modelo estético es superior, por ejemplo, a la fricción.

Gerfried Stocker: La fricción también es interesante. Pero es una cuestión de los diferentes niveles de escritura e invención del código, de creación de una interfaz como tecnología o de la interfaz en un sentido literal, donde puede tener lugar la comunicación entre el proyecto y el resultado. Esta integración es muy importante. En el caso de Golan Levin es evidente que realmente no crea fricción: crea unas imágenes preciosas. Pero ya he dicho que las imágenes no son importantes y, de hecho, tampoco lo es la fricción. Y por este motivo no acepto una definición de software artístico que solo se refiera a los virus, los códigos y a este potencial subversivo del software artístico; y tampoco aceptaría ninguna descripción de software artístico que solo hiciera referencia a imágenes y sonidos bonitos. La calidad se encuentra en un área muy distinta y corremos el riesgo de investigar en la dirección equivocada.

Rafael Lozano-Hemmer: Esas nociones no son mutuamente excluyentes. Por una parte, cuando hablamos de software artístico de forma muy fetichista, como lo hace Gerfried Stocker, comparto el tipo de atracción formal por el fetiche del código. A la vez, debemos ser conscientes de que el código no está en absoluto no socializado. En otras palabras, el código, como sabemos los que programamos en distintos lenguajes, es en sí mismo una cierta colaboración con las personas que lo escribieron. De forma que, en la medida en que no hay pureza en este tipo de expresión, me interesa el hecho de que el propio código es social. Y podemos decir lo contrario en relación con otras obras, como por ejemplo las mías, que son montajes con otro tipo de medios o modelos sociales. En estos casos también trabajamos con un código, aunque no sea un código informático; puede ser un código lingüístico, político, social, económico o del tipo que sea. Creo que la obra de Golan Levin es extremadamente social; la forma en que reacciona mi cuerpo ante su código, como resultado de la elección de imágenes y sonidos y del efecto buscado, es extremadamente social. A mi entender, esto es lo que tiene valor, tanto si lo hizo con el programa Director como de otra forma; creo que tiene mérito y es interesante.

Gerfried Stocker: Estoy de acuerdo. Este es el aspecto importante de la interfaz, que forma parte de este cambio. Y creo que sería un error separar estas dos cosas. Solo para aclararlo de nuevo, me estaba refiriendo concretamente al aspecto de la estética. ¿Qué sería la estética en este tipo de arte? Afortunadamente hay otros criterios para abordar el arte que no son solo estéticos. Y el software artístico también debe referirse a estos otros criterios.

Anne Nigten: Creo que hay una diferencia entre código artístico y software artístico o *artware*. Creo que no debemos confundirlos. Personalmente no estoy demasiado interesada en la estética del código, sino en lo que se crea mediante este código o en lo que este código pueda aportar. Por lo tanto, no me refiero a la idea convencional de arte, sino a una experiencia más amplia producida por el código.

Daniela Plewe: Sigo planteándome la noción de transparencia. ¿Cómo podemos comunicar al público, al espectador, algo tan complejo en lo que todos estamos interesados? ¿Queremos realmente esto o nos gusta tener un público muy especializado? Como dice Gellernter: "la estética de la máquina y el algoritmo puro optimizado". Si finalmente el matemático dice lo que está bien y qué es arte, entonces abrimos todo el espectro de lo que estamos hablando a las matemáticas y a su belleza. Ya que hay un vacío en la definición de arte, creo que incluso estaría de acuerdo en que en un futuro se incluyera a las matemáticas.

Enlaces relacionados:

➡ Festival Transmediale:

<http://www.transmediale.de>

➡ Declaración del jurado de Transmediale sobre el software artístico:

<http://www.transmediale.de/en/02/softjurystate.php>

Cita recomendada

BROECKMANN, Andreas (2003). "¿El software artístico es material artístico genuino?". *Artnodes*, n.º 2 [artículo en línea].

DOI: <http://dx.doi.org/10.7238/a.v0i2.683>

Instituciones colaboradoras:



<http://www.n5m.org>

transmediale

<http://www.transmediale.de>

Fecha de publicación: diciembre de 2002