

titulo:

flaxus

obra[sintesis]:

Flaxus es un software desarrollado para realizar performances visuales en tiempo real bajo el manifiesto TOPLAP[<http://www.toplap.org>]. Donde la pieza grafica es generada por codigo en el momento mismo de su ejecucion, acercando la experiencia artistica al hecho performante de la musica o la danza. Flaxus incorpora y plantea a su vez un nuevo paradigma. Permite que un ejecutor produzca visuales de acuerdo a algo que esta in situ, pero que cientos de otros participantes pueden ver activado por audios individuales en otros rincones del planeta. De esta manera se invierte el valor habitual estetico. Una misma composicion visual se vuelve reactiva a la perpeccion musical individual. Promoviendo tambien el trabajo en redes, Flaxus es una herramienta operable colaborativamente, permitiendo la realizacion en tiempo real de una pieza entre distintos ejecutores a traves de una conexion a internet.

motivo:

Creemos que las artes visuales electronicas necesitan aun mucha mas experimentacion y profundizacion en su busqueda elemental. Flaxus es un trabajo de campo, un postulado. La anteposicion de la performance visual a la musical. El uso de las redes para crear elementos reactivos visuales capaces de reaccionar a distintos entornos en simultaneo. La experiencia de lo colaborativo a distancia. La programacion de codigos y procesos interpretativos en tiempo real. Buscamos con esta pieza inspeccionar los limites de la experiencia visual en vivo.

camino:

Hace un tiempo nos propusimos realizar un camino de investigacion en las artes visuales electronicas explorando la capacidad performante de estas en vivo. Buscábamos encontrar alguna semejanza en el acto de tocar un instrumento musical en vivo, y la generacion de contenidos visuales.

Dimos en un principio con el acto de Visual Jockey o VJ. Un acto performante visual en el que un operador lanza imagenes que acompañen al estímulo auditivo.

Encontramos en el acto de VJing algo mas similar al del DJ que no compone sino que mezcla. Aunque esta mezcla se realice con material de su autoria no era eso lo que buscábamos.

Existe una diferencia radical en el acto de tocar una guitarra nota a nota en vivo, para componer la melodía, al de determinar en que momento algo se debe oír, aunque en su momento, con anterioridad, uno lo haya grabado nota a nota.

Nos ocupaba el mismo planteo el acto de VJ. La performance no era 100% live como puede serlo tocar el piano.

Determinamos que era importante entonces constituir una unidad minima visual. Asi como la nota en la musica.

El inconveniente es que la musica tiene un solo eje de datos. La nota es una onda que avanza en una unica direccion, con valores distintos en el tiempo. La imagen contiene mucho mas datos en su relacion a lo digital. Por ejemplo tomando la pantalla que es plana como ejemplo, esta tiene 2 ejes cartesianos que crean la grilla de pixeles a diferencia del unico eje del audio.

A eso sumamos la dificultad de la relacion entre ese plano y una fraccion de tiempo visual que en su parametro estetico acorde a una fluides optima podriamos hacer oscilar entre 1/30 partes de segundo a 1/12 partes de un segundo.

Vimos entonces que la composicion tenia que darse por otro lado y no por una metaforica translacion del proceso de creacion de la onda a la imagen que solo daba por construido distintos valores de un unico color que envolvia todo el plano o alguna variante mas compleja que no satisfacía nuestras expectativas.

[Al querer constituir una translacion metaforica, y convertir la imagen en un proceso similar al de la musica, solo podiamos determinar en tiempo real datos minimos en la linea de tiempo, como on/off de ciertos elementos, o graduaciones de color u forma]

Encontramos entonces una relacion indirecta entre el hecho de tocar un instrumento con el de escribir codigo de programacion.

Investigando este punto dimos con el manifiesto TOPLAP.

Este manifiesto pertenece a una corriente de origen musical que hoy tiene ya un brazo visual.

Dicho manifiesto expone la misma problematica e interes en las expectativas de performance. Y lo resuelve en el mismo punto al que nosotros llegamos.

Plantea una plataforma donde el contenido visual es programado (Definimos como programar la escritura y estructuracion de codigo entendible por la maquina y que esta traduce en imagenes). De esta forma el performante se ve obligado a actuar constantemente para mantener una fluides estetica en la imagen.

Esta formulacion posee mayores semejanzas con el acto de tocar en vivo musica, y es quizas una vuelta

atrás a las consignas que proponían personas como Varesse en los orígenes del acto musical electrónico. En dichos orígenes uno de los intereses era el de conseguir una performance que sea repetible de forma idéntica por medios electrónicos. Dado que la ejecución musical en aquellas épocas dependía de muchos factores (músicos, lugar, director de orquesta, público, etc.). Muchos eran los factores que determinaban la representación de la obra final.

Un poco nuestro interés es retomar ese factor de error puro que es propio de la ejecución de una pieza en el momento. Lo efímero de la ejecución artística en sí.

El manifiesto TOPLAP se suscribe a estos requisitos también, donde la pieza es efímera. El código producido no queda guardado y se pierde tras su ejecución.

Otra premisa del manifiesto es que el código sea visible para el público, así como las manos del guitarrista en su exquisito movimiento hacen parte mismo del acto más allá de lo incomprensible de estos vaivenes de los dedos.

Nos propusimos entonces crear un soft que corresponda a este manifiesto en todos sus principios. No sería este el primer software, inclusive el nombre seleccionado lo demuestra.

Existe un software previo, de nombre Fluxus haciendo memoria del movimiento artístico homónimo.

Flaxus, el nombre elegido por nosotros, hace uso del mismo estigma, pero reemplazando la sílaba inicial por "Fla" reemplazo habitual en el software y en la web cuando el programa tiene algún tipo de relación con el Flash. Que es el software en el que desarrollamos la aplicación.

Pero no nos detuvimos ahí, nos propusimos entonces plantear un paso más con nuestra herramienta.

Incorporar la realidad de las redes de hoy. El concepto y descomposición atómica que se puede realizar a través de internet.

Nuestro software está constituido sobre un replanteo de la ejecución visual en vivo.

Este planteo es la posibilidad de que una persona esté generando esquemas visuales, componiendo en vivo en un rincón del planeta, y que otra persona pueda verlo en vivo en otra parte.

Pero no como transmisión pasiva. Entiéndase como transmisión pasiva el paso del dato entero de un punto al otro sin deformaciones mayores que la pérdida de calidad consiguiente a la transmisión misma.

Por ejemplo la transmisión por televisión o radio.

El software tiene como "core" o cuerpo forjador una serie de elementos reactivos a estímulos auditivos.

De esta forma la composición visual se ve alterada y entra en juego con la música aplicada a él.

Nuestro replanteo es el de imponer como motor conductor del ejercicio audiovisual la imagen.

Permitiendo que cada usuario pueda aplicarle la música que quiera.

De esta forma si un ejecutor está en una parte del mundo escribiendo código en vivo, en cada lugar que este código sea reinterpretado, lo hará reaccionando a la música o sonido particular provistos en el lugar y el momento independientemente de lo que ocurra en cada otro punto donde sea visto.

Entonces la composición estética visual deja de ser estricta y se vuelve flexible a la circunstancia.

Pero esta flexibilidad no se limita a la transmisión lineal de la comunicación en la que hay un emisor y un receptor. Los canales digitales permiten una performance circular, en sus canales de transmisión.

Lo que contempla el flaxus es la colaboratividad de la experiencia performante.

No existe un performante único, sino que el espectador puede volverse performante y modificar la pieza hacia afuera, hacia el resto de los receptores, posibles emisores en cualquier momento.

La performance pasa a ser un diálogo en conjunto entre muchas partes, el paradigma de performante público se disuelve, y puede hasta invertirse por momentos.

Así logramos ese error estético posible como base misma del sistema. En cada lugar o momento puede ser completamente distinta la percepción final de la ejecución sin alterar en nada la composición artística de la pieza.

gramática:

La estructura lógica de este lenguaje de programación está basada en un sistema de jerarquías modulares. Estas están insertadas una dentro de otra y se resuelven de adentro hacia afuera, despejando incógnitas de la misma manera que se hace en la matemática.

Cada estructura por lo general tiene como primer elemento un verbo y como segundo un sustantivo sobre el cual opera el verbo seguido en algunos casos de distintos adjetivos o verbos secundarios que operan sobre el sustantivo)

Estas estructuras van siempre entre parentesis.

Un ejemplo de verbo es:

(Build_Cube)

que crea un cubo

(deleteModel ACTIVE_MODEL)

tiene un verbo y un sustantivo

esta construcción dice que borre el modelo activo, o sea el último modelo creado o accedido.

(Build_Cube 30)

crea un cubo con escala 30.

En este caso no hay sustantivo porque es tácito

(setVar miCubo ACTIVE_MODEL)

aquí estamos denominando un nuevo sustantivo miCubo con el valor resuelto de un adjetivo

Existe otro tipo de estructura que hace referencia a un grupo y en general su primer valor es un elemento de ese grupo
(math random)
en este caso toma el valor random del elemento math

Las estructuras a su vez pueden reemplazar valores dentro de otra estructura
(setPosition (math random 100) 0 0)
el verbo set position opera automáticamente sobre el último modelo creado, cambiando su posición con los 3 adjetivos o parámetros ingresados representando estos X Y Z respectivamente.
En el ejemplo vemos que el valor de X es resuelto por un valor random de 0 a 100.

Las estructuras entonces pueden servir no solo como acción determinada, sino como definición de un valor para otra acción de mayor jerarquía.

capacidades:

El Flaxus es una herramienta conceptual experimental. No es un software de alto rendimiento. Tiene un gran valor educativo así como lo tuvo en su momento el software Design By Numbers [<http://dbn.media.mit.edu/>] creado originariamente por Jhon Maeda en el MIT o como lo puede tener hoy el Processing [<http://www.processing.org>] creado por Ben Fry (Broad Institute) y Casey Reas (UCLA Design Media Arts).

La herramienta permite crear polígonos simples en tres dimensiones en tiempo real en un espacio reactivo al sonido.

Incorpora el uso de partículas en dos y tres dimensiones.

Manejo de variables y cálculos matemáticos

Creación de bitmaps en tiempo real

Distintas operaciones de blending de layer.

Uso de texturas, tipografías y gradients generados en tiempo real.

Operaciones directas sobre video y bitmap, esto es Kerneling, Convolution Filters, Bitmap Copy, Video Feedback, etc

Siendo todas estos procesos ejecutados en tiempo real.

A esto se añade que la herramienta es colaborativa, lo que permite a muchas personas operar sobre la misma pieza en simultáneo y compartir un código y su ejecución al instante.

tecnología:

El software entero está construido con Adobe Flash 8. Que a pesar de haber comenzado en sus orígenes como una herramienta exclusiva para la construcción de animaciones hoy tiene un potente lenguaje orientado a la programación estética. El cual por sobre todo nos permite realizar en tiempo real operaciones sobre bitmap y ser portable a cualquier plataforma que soporte el Player de Adobe Flash 8. De esta forma en un único desarrollo pudimos tener un software que puede correr embebido en una página web y que sea multiplataforma.

que sigue:

En principio nuestro próximo plan es aumentar la documentación y agregar ejemplos.

Creemos que el análisis de ejemplos es una de las formas más robustas y eficaces en la comprensión de una herramienta.

El software aún está en fase beta, así que de aquí en adelante estaremos corrigiendo errores que obtendremos de la misma comunidad de usuarios.

Nuestro próximo acto será implementar un sistema de foros para que la comunidad pueda postear preguntas, códigos o scripts

comunidad:

El software es lo ponemos a disposición de la comunidad bajo la Licencia GNU GPL [<http://creativecommons.org/licenses/GPL/2.0/>]

Compartiendo el código de este por si alguien quiere modificarlo o corregirlo, así como su libre uso por cualquier individuo

Solicitando solamente que queden citadas las fuentes y autores de la herramienta

No está permitido el uso comercial de esta plataforma

screens:



artistas:

Ivan Ivanoff
Autodidacta
www.i2off.org
avatar@i2off.org

Jose Jimenez
Autodidacta
www.r3nder.net
noremorse@gmail.com