



Tres Estudios Abiertos

Nuevas prácticas performáticas audiovisuales experimentales para el navegador.

Emilio Ocelotl

5 de abril de 2021

Resumen

Tres Estudios Abiertos es un proyecto de investigación que busca adentrarse en nuevas prácticas experimentales y audiovisuales para el navegador. Parte de la programación de un esqueleto de granulación audiovisual complementado con módulos personalizados de software. Atiende a niveles altos de programación para el control de software por medio de una interfaz de texto.

Toma en cuenta piezas para el navegador que fungan casos de estudio. El proyecto busca demostrar que las lógicas de los lenguajes de programación posibilitan formas de pensamiento específicos con consecuencias estéticas definidas.

Los casos de estudio son entramados de audio y pretenden ser ligeros, sin instalaciones, sin referencias a instalación de dependencias de terceros, basados en la web, distribuidos y optimizados para el bajo consumo de recursos de la computadora.

Hasta el momento considera a *Javascript*¹ como el lenguaje principal de los casos de estudio. De manera secundaria el proyecto se pregunta si es posible deducir ideas estéticas de las arquitecturas de los lenguajes de programación para la escritura de interfaces textuales de control.

Adicionalmente al estético y tecnológico, consideramos un tercer hilo que corre paralelamente al bucle de investigación-creación: el investigativo. En esta pista están presentes reflexiones e implicaciones políticas en la escritura de y con software.

La investigación busca relacionar la escritura del texto resultante con la escritura de software y otros medios (principalmente repositorios de código pero también imágenes, sitios web, sonido). De esta manera el documento resultante busca desbordarse del papel y de la palabra escrita.

Palabras clave— software, navegador, javascript, granulación, cyberespacio, par a par, programación estética, distribución, webAssembly, WebGL

Antecedentes

Los antecedentes de este proyecto describen la transición del desarrollo de software para la realización de sistemas interactivos a la escritura de módulos de software audiovisuales. Estas experiencias parten de la optimización y la ligereza del lado del hardware (por ejemplo, con el uso de computadoras de placa reducida como Raspberry Pi o Jetson Nano) y de sistemas ligeros, con pocas capas de abstracción y accesibles de síntesis y renderización de audio y video en el navegador.

Parte de los antecedentes están directamente relacionados con la experiencia performática de escribir de código al vuelo con fines creativos, audiovisuales y experimentales, de manera similar a como lo describen Villaseñor-Ramírez y Paz (2020) para los casos de Barcelona y Ciudad de México.

De manera directa, los siguientes proyectos están relacionados con la investigación y se convierten en casos de estudio, en algunas otras ocasiones funcionan como antecedentes directos:

- Caso de estudio. THREE.studies (EmilioOcelotl, 2020c) y en específico, las iteraciones threecln (EmilioOcelotl, 2020b) y threeBEASTs (EmilioOcelotl, 2021b)
- Caso de estudio. Anti (EmilioOcelotl, 2020a)
- Antecedente directo. Diálogo IV @ Coloquio Salvador Contreras. Artes Sonoras y Creación Musical en México: Siglo XXI (EmilioOcelotl, 2021a)
- Antecedente directo. Panorama. Escritura de espacios libres e inmersivos para el performance audiovisual (PiranhaLab, 2020)²

Algunos proyectos similares al que describe *Tres Estudios Abiertos* son:

- **Nivel Bajo:** Ruffbox (the-drunk coder, 2019), WebAssembly/Rust Tutorial³ y Flocking (continuing creativity, 2011).
- **Nivel Medio:** (khilnani, 2012), Web Audio API⁴, Tone.js⁵ y supercolliderjs⁶.

- **Nivel Alto:** Estuary (dktr0, 2019), Troop (Qirky, 2016), flock (munshkr, 2019), tilt (munshkr, 2021), LiveLab (ojack, 2020b), Hydra (ojack, 2020a), timeNot (AFrancoB, 2019) y seis8s (luisnavarrodelangel, 2020)

Marco Teórico

La presente investigación parte del giro de los nuevos medios y de los estudios del software (Manovich, 2001). Como una extensión del punto de partida, la investigación se adscribe a la escritura con y sobre software (Soon y Cox, 2020). *Tres Estudios Abiertos* desemboca en el papel que juega la experiencia subjetiva y las implicaciones políticas y sociales en la programación estética (Cox y McLean, 2012).

Plantamiento del problema

De entre los proyectos similares destacamos aquellos que son de Nivel Alto para responder a la pregunta: ¿Cuál es la diferencia entre los proyectos mencionados y *Tres Estudios Abiertos*? Dos perspectivas podrían aclarar el punto de partida del proyecto. La primera es funcional y hace referencia a la solución de problemas partiendo de una comunidad que ejecuta, retroalimenta y enriquece al proyecto. La segunda, apuesta por la diversidad en el desarrollo de interfaces de control y que se manifiesta de una manera muy específica en el proyecto Estuary. El presente proyecto busca responder en un momento anterior a la realización de módulos de software si hay diferencias estéticas heredadas de notaciones musicales y computacionales, lenguajes de programación, estilos musicales, flujos de voltaje que desembocan en síntesis de audio / imagen e incluso planteamientos críticos sobre decolonialidad. Referimos a este conjunto de diferencias como decisiones de diseño en sintaxis de control que tienen consecuencias en la estética que resulta de controlar motores de audio y video. En este sentido, *Tres Estudios Abiertos* es una búsqueda que orbita en estas decisiones y se adscribe a la diversidad en la escritura de y con software.

Premisa/Hipótesis

Las lógicas de los lenguajes de programación posibilitan formas de pensamiento específicos y distintos entre sí en la conformación de software orientado al performance audiovisual y a la programación de obras fijas.⁷

Metodología

¿Cómo responder a la hipótesis principal? ¿Es posible medir de alguna manera la experiencia estética y subjetiva de un lenguaje de programación? ¿Es posible encontrar una respuesta en el alto nivel?

Justificación

Problematizar el software desde lo político y en la escalada de recursos para ejecutar un programa. La escritura de programas de bajo nivel como una declaratoria de motivaciones adicionales a la resolución funcional de un programa.

El punto de partida puede ser la escritura de software para la investigación.

Esto puede implicar aspectos relacionados con uso de pocos recursos energéticos y de hardware. La discusión puede llevarse hacia la inclusión / exclusión de experiencias. De manera indirecta puede plantearse la crítica a la industria del software como los mayores escaladores de recursos.

Objetivo principal

Escribir una serie de módulos interconectados de software para realizar síntesis granular audiovisual en el navegador.

Objetivos Secundarios

- Implementar el sistema en obras. Detonador de la relación entre práctica y reflexión escrita de cara a la tesis.

- Utilizar una serie de instrucciones personalizadas como interfaz textual que puedan mediar entre niveles y que discutan con el contexto cultural en el que se enmarcan.
- Realizar reflexiones a manera de documentación, teniendo como eje de vinculación conceptos tecno-sociales y estéticos.
- Aportar módulos de desarrollo tecnológico que puedan ser de utilidad para otros proyectos.
- Estrategias de colaboración y creación a distancia, por medio de la computadora y e instrumentos acústicos.
- Comparar críticamente el desarrollo tecnológico y la tecnología existente.
- Documentar los desarrollos y resultados secundarios realizados en colectivo.

Estructura sugerida

La estructura es resultado de la discusión que surge entre la investigación convencional, la investigación crítica y la investigación artística.

- **Resumen.**
- **Introducción.** Antecedentes, motivaciones e hipótesis/premisa (¿Qué?).
- **Metodología.** ¿Cómo se responde a la hipótesis/premisa?
- **Resultados.** Casos de estudio, descripción de piezas y entramados de código.
- **Conclusión.** ¿Se demostró la hipótesis/premisa?

Notas

¹¿Todavía es posible usar otros lenguajes de programación?

²Una versión en construcción y en línea del artículo se puede encontrar en <https://piranhalab.github.io/panorama/>

³<https://www.toptal.com/webassembly/webassembly-rust-tutorial-web-audio> (Consultado el 5 de abril de 2021)

⁴https://developer.mozilla.org/es/docs/Web/API/Web_Audio_API

⁵<https://tonejs.github.io/>

⁶<https://github.com/crucialfelix/supercolliderjs/>

⁷Hipótesis premisa que traspola la propuesta que Ludwig Wittgenstein expresa en el *Tractatus Logico Philosophicus*: ¿Los límites de mi lenguaje (de programación) son los límites de mi mundo?

Referencias

AFrancoB (2019). timenot. <https://github.com/AFrancoB/timeNot>.

continuing creativity (2011). Flocking. <https://github.com/continuing-creativity/Flocking/>.

Cox, G. y McLean, A. (2012). *Speaking Code: Coding as Aesthetic and Political Expression*. The MIT Press.

dktr0 (2019). Estuary. <https://github.com/dktr0/estuary>.

EmilioOcelotl (2020a). anti. <https://github.com/EmilioOcelotl/anti>.

EmilioOcelotl (2020b). threecln. <https://github.com/EmilioOcelotl/THREE.studies/tree/main/threecln>.

EmilioOcelotl (2020c). Three.studies. <https://github.com/EmilioOcelotl/THREE.studies>.

EmilioOcelotl (2021a). Dialogoiv. <https://github.com/EmilioOcelotl/dialogoIV>.

EmilioOcelotl (2021b). threebeasts. <https://github.com/EmilioOcelotl/THREE.studies/tree/main/threeBEASTs>.

khilnani (2012). supercollider.web. <https://github.com/khilnani/supercollider.web/>.

luisnavarrodelangel (2020). seis8s. <https://github.com/luisnavarrodelangel/seis8s>.

Manovich, L. (2001). *The Language of New Media*. Leonardo (Series) (Cambridge, Mass.). MIT Press.

munshkr (2019). flok. <https://github.com/munshkr/flok>.

munshkr (2021). tilt. <https://github.com/munshkr/tilt>.

ojack (2020a). hydra. <https://github.com/ojack/hydra>.

ojack (2020b). Livelab. <https://github.com/ojack/LiveLab>.

PiranhaLab (2020). Panorama. <https://github.com/piranhahalab/panorama>.

Qirky (2016). Troop. <https://github.com/Qirky/Troop>.

Soon, W. y Cox, G. (2020). *Aesthetic Programming: A Handbook of Software Studies*. Open Humanities Press.

the-drunk coder (2019). ruffbox. <https://github.com/the-drunk-coder/ruffbox>.

Villaseñor-Ramírez, H. y Paz, I. (2020). Live Coding From Scratch: The Cases of Practice in Mexico City and Barcelona. En *Proceedings of the 2020 International Conference on Live Coding (ICLC2020)*, pp. 59–68, Limerick, Ireland. University of Limerick.