

Lenguaje de programación visual como recurso creativo en el ámbito de las bellas artes: El caso de VVVV, a multipurpose toolkit¹

(Visual Programming as a Creative Resource in the Field of Fine Arts: The Case of VVVV, a Multipurpose Toolkit)

José Antonio Vertedor–Romero,² Universidad de Málaga, España
Isidro López–Aparicio Pérez, Universidad de Granada, España

Resumen: Este artículo se enmarca en una línea de investigación en torno a prácticas artísticas del ámbito de las Bellas Artes en las que se trabaja con procesos computacionales y código de programación visual como recursos creativos. Las nuevas tecnologías amplían el espectro interdisciplinar y los recursos digitales generados constituyen un reto para las disciplinas artísticas que requieren permanecer actualizadas en cuanto a los recursos de su presente. Desde nuestra experiencia artística como productores artísticos, consideramos imprescindible el diseño y desarrollo de nuevas metodologías que impliquen el conocimiento de las diferentes herramientas digitales que encontramos en la actualidad a nuestra disposición. Además, entendemos la necesidad de acercarlas al alumnado con el objetivo de dar a conocer sus posibilidades creativas para su utilización en diferentes propuestas artísticas sin necesidad de tener una especialidad en ingeniería informática. Podemos encontrar numerosos recursos digitales para este propósito que pueden ser los programas, Max o TouchDesigner entre otros. Nos centramos con esta propuesta en la herramienta multipropósito VVVV por ser un programa cuyo origen ha sido definido por sus desarrolladores como un recurso pensado para artistas con el que no se necesita a priori tener un conocimiento en código de programación textual para poder operar con esta herramienta. Este programa concede la posibilidad de generar programas específicos y trabajar en entornos interdisciplinares híbridos con recursos para inteligencia artificial, procesamiento de imagen y sonido o diseño de Motion Graphics entre otras tendencias artísticas contemporáneas.

Palabras clave: creatividad computacional, lenguajes de programación visual, VVVV: a multipurpose toolkit, creación audiovisual, live coding, arte digital

Abstract: This article is part of a line of research into artistic practices in the field of Fine Arts in which computational processes and visual programming code are used as creative resources. New technologies broaden the interdisciplinary spectrum, and the digital resources generated constitute a challenge for artistic disciplines that need to remain up to date with the help of the present. From our creative experience as artistic producers, we consider it essential to design and develop new methodologies involving knowledge of the current digital tools. Furthermore, we understand the need to bring them closer to the students to make them aware of their creative possibilities for use in different artistic proposals without specialising in computer engineering. We can find numerous digital resources, such as the software Max or TouchDesigner. We focus with this proposal on the multipurpose tool VVVV because it is a programme whose origin has been defined by its developers as a resource designed for artists with which it is not necessary a priori to know textual programming code to be able to operate with this tool. This programme offers the possibility of generating specific programmes and working in hybrid interdisciplinary environments with resources for artificial intelligence, image and sound processing or Motion Graphics design, among other contemporary artistic trends.

Keywords: Computational Creativity, Visual Programming Languages, VVVV: a Multipurpose Toolkit, Audiovisual Creation, Live Coding, Digital Art

¹ Lenguaje de programación visual creado para facilitar el prototipado y desarrollo de interfaces <https://vvvv.org/>

² Autor de Correspondencia: José Antonio Vertedor-Romero, Plaza de El Ejido, 1, Departamento de Arte y arquitectura, Universidad de Málaga, Málaga, España, 29013, España. email: vertedor@uma.es

Procesos Creativos Computacionales en el ámbito artístico

El campo de las Bellas Artes siempre se ha debatido en una constante lucha entre la innovación y tradición. Encontramos un claro ejemplo de esto en las personas que protagonizaron la época de las vanguardias artísticas, quienes desde un pensamiento multidisciplinar nos mostraron una rebeldía que buscaba romper esquemas, sorprender o escandalizar al espectador a través de diferentes procesos creativos con el objetivo de ofrecer una visión del mundo desde una visión experimental. Por otro lado, vemos que el academicismo artístico se centra en encontrar las reglas que definan qué es *buen arte*. Su objetivo es que éste sea constructivo para el legado cultural de la humanidad. Como nos ha mostrado la Historia del Arte, sabemos que hasta finales del siglo XIX los avances tecnológicos no tuvieron gran impacto en la definición de las artes tradicionales, como son la Arquitectura, Escultura, Pintura, Música, Poesía e Interpretación. La tecnología creó nuevas técnicas o desviaciones, como ocurrió con el caso de la fotografía, que podría ser considerada como una *desviación* de la pintura por compartir principios muy similares en cuanto a diseño y composición (Benjamin 1982).

El ámbito de la creación artística se ha constituido como un espacio inter- y multidisciplinar en el que se manifiesta la relación Arte-Ciencia-Tecnología. Existen numerosos ejemplos de esto en el entorno creativo, como ocurre con la tendencia artística surgida en la década de los años sesenta denominada *Computer Art* o la actual *Data Art*, por ejemplo. En la década de 1950, muchos artistas y diseñadores trabajaban con dispositivos mecánicos y computadoras analógicas. Esta tendencia puede entenderse como precursora del trabajo de los artistas digitales pioneros que la siguieron. Cabe señalar en este contexto una de las primeras obras electrónicas de la colección de V&A³, *Oscillon 40*, que data de 1952. En esta pieza, el artista, Ben Laposky, utilizó un osciloscopio para manipular las ondas electrónicas que aparecían en la pequeña pantalla fluorescente.

A principios de los años sesenta, las computadoras se encontraban principalmente en instalaciones académicas o de investigación con sus propios laboratorios informáticos. El funcionamiento de un laboratorio de computadoras centrales requería un conjunto de personal técnico para hacer funcionar las diferentes unidades que componían estas máquinas. En consecuencia, las primeras personas que crearon *Computer Art* no eran artistas, sino matemáticos, científicos y programadores que tenían acceso a estas máquinas. Una de las primeras exposiciones de computadoras en Norteamérica tuvo lugar en la Howard Wise Gallery de Nueva York en 1965. En ella se expusieron obras de dos investigadores de los Laboratorios Bell, Michael A. Noll y Bela Julesz. Noll y Julesz trabajaron con el IBM 7094 y un trazador de microfilmes SC-4020. La muestra en sí no fue bien recibida por la crítica de arte, que llegó a definirla en estas palabras:

Hasta ahora los medios son de mayor interés que el fin. Esta colaboración revolucionaria da como resultado patrones geométricos sombríos y muy complejos que excluyen el más mínimo ingrediente de sensibilidad manual. Independientemente de lo que nos depare el futuro –los científicos predicen una época en la que casi cualquier tipo de pintura podrá ser generada por ordenador– el toque real del artista ya no desempeñará ningún papel en la realización de una obra de arte⁴ (Preston 1965, 23).

³ <https://collections.vam.ac.uk/item/O187634/oscillon-40-photograph-laposky-ben/>

⁴ Traducción de los autores del texto original: [So far, the means are of greater interest than the end, this revolutionary collaboration resulting in bleak, very complex geometrical patterns excluding the smallest ingredient of manual

Lejos de esta afirmación realizada por Stuart Preston en el diario, *The New York Times*, en el año 1965, la realidad es que el *software* está cada vez más vinculado a nuestra vida en todos los aspectos: económico, cultural, creativo y político. Esto se produce de una manera evidente a la vez que invisible. El teórico de los nuevos medios, Lev Manovich, afirmaba en su trabajo, *Software takes command*, que se ha escrito mucha literatura en torno al uso que se hace del *software*, además de lo que permite hacer, pero que esto ha sido abordado desde un pensamiento principalmente técnico. Sin embargo, cada vez más, artistas, investigadores, ingenieros, hackers, diseñadores y académicos de las humanidades y las ciencias sociales se ponen de acuerdo en que es “necesario expandir el conocimiento sobre el *software* para abordar la complejidad de algunas de las acciones que pretenden llevar a cabo” (Manovich 2013, 14).

Cabe señalar en este contexto la figura del artista-programador, la cual desarrolla su obra artística a través de la experimentación con código de programación. El artista-programador se apoya en el lenguaje informático, lo que le confiere una ventaja única sobre la interacción entre el ser humano y la computadora en un contexto creativo. Encontramos una relación directa con lo que la Historia del Arte nos ha mostrado en toda su trayectoria, es decir, el ser humano ha utilizado desde su origen los diferentes recursos y lenguajes para expresarse de manera artística (Gombrich 2008). En consecuencia, podemos afirmar la existencia de una profunda relación entre los lenguajes informáticos y la expresión creativa. Se hace patente de esta manera que estos campos del pensamiento se apoyan mutuamente, y que el artista-programador puede comprometerse plenamente con ambos a través de un proceso creativo multi- e interdisciplinar.

Ubicamos al artista-programador dentro del contexto de las artes, concretamente en el ámbito de la *Computer Art*, al tiempo que se refiere a la identidad singular del programador como artista. Este sujeto se compromete con las relaciones humanas internas mediante un trabajo que se organiza entre la percepción, la cognición y la computación. Esto está relacionado con el funcionamiento de los algoritmos con los que opera. Los procesos creativos son bastante más misteriosos que los procesos informáticos, de los que podemos extraer que están bien definidos, son predecibles y están sometidos al control humano. Sin embargo, “para los artistas-programadores, el proceso informático es el agente principal en su proceso creativo” (McLean 2011, 118). De lo expuesto hasta aquí podemos extraer que la necesidad de conocer lenguajes de programación para su uso en la creación artística computacional es primordial en la actualidad. Además, se pone de manifiesto la necesidad de que estos sean accesibles a cualquier persona, aunque no esté familiarizada con la informática.

Pueden encontrarse en el mercado numerosos programas que utilizan el lenguaje de programación visual, como, por ejemplo, Max, Pure Data, TouchDesigner o VVVV. Estos programas son utilizados habitualmente por artistas, productores audiovisuales y creadores de música electrónica para introducir diferentes elementos de interactividad en sus piezas, actuaciones o instalaciones. Uno de los motivos principales de la popularidad de los lenguajes de programación visual es, sin duda, su aparente sencillez para la programación, es decir, al ser un lenguaje gráfico parece no implicar más que la conexión de varios objetos entre sí. Sin embargo, crear algo interesante con estas herramientas es a menudo más difícil de lo que esa primera impresión podría implicar. Tras aprender algunos conceptos iniciales, como el ordenamiento de los mensajes operadores, el siguiente paso importante es “adquirir conocimientos básicos sobre los objetos más importantes y sobre cómo conectarlos para formar patrones de uso común” (Hadjakos et al. 2015, 1).

Nuestro proceso de investigación nos dirige a través de diferentes capas de representación, empezando por los símbolos, luego palabras, el lenguaje y la notación, para considerar el papel que estas representaciones pueden desempeñar en la creatividad humana. Formamos una

sensibility. No matter what the future holds –and scientists predict a time when almost any kind of painting can be computer generated– the actual touch of the artist will no longer play any part in the making of a work of art].

perspectiva interdisciplinar desde la psicología, la informática, la lingüística, la interacción persona-ordenador, la creatividad computacional, la tecnología musical y las artes. Desarrollamos y demostramos el potencial de este punto de vista para potenciar la experiencia artística a través de la introducción práctica de creación de prototipos de *software*, obras de arte o actuaciones. En particular, nos centramos en la creación de obras que potencian el papel de la percepción en la semántica simbólica, es decir, la incorporación de la representación del tiempo en el lenguaje de programación partiendo de la visión espacial en la improvisación y la experiencia del arte. Vinculados a este contexto podemos encontrar diferentes agentes creadores. Destacaremos en este caso a dos de los productores más relevantes a nivel mundial en la línea de investigación de la producción audiovisual computacional, tanto por su amplia proyección internacional como por su gran experiencia en este ámbito de estudio. Los casos de Daito Manabe y Ryoji Ikeda.

Daito Manabe, diseñador de interacción, programador y DJ, cuya sede se encuentra en Tokio, es uno de los artistas transdisciplinares más importantes del siglo XXI. Ha sido considerado por la empresa estadounidense Apple “uno de los 30 creadores de referencia del panorama actual” (Arday). Desde 2015, ha trabajado junto a Motoi Ishibashi como codirector de *Rhizomatiks Research*, la división de la firma dedicada a explorar nuevas posibilidades en el ámbito de la expresión técnica y artística, centrándose en proyectos intensivos de I+D. El trabajo de Manabe abarca sobre todo diseño, arte y entretenimiento. El autor adopta un nuevo enfoque de los materiales y los fenómenos cotidianos. Sin embargo, su objetivo final no es simplemente el de obtener un realismo rico y de alta definición, más bien, su práctica se basa en una cuidadosa observación desde la que descubrir y dilucidar las potencialidades esenciales inherentes al cuerpo humano, los datos, la programación, las computadoras y otros fenómenos, explorando así las “interrelaciones y los límites entre lo analógico y lo digital, lo real y lo virtual” (Manabe).

Tomaremos como ejemplo su obra *Morphecore*, del año 2020. Se trata de un buen ejemplo con el que el autor pone de manifiesto la hibridación entre arte, ciencia y tecnología. Para este trabajo, el artista japonés parte de la investigación del neurocientífico, Tomoyasu Horikawa, en la que se presenta una técnica de *Brain Decoding* con la que se analizan los patrones de la actividad del cerebro humano para leer estados mentales e incluso sueños (Horikawa y Kamitani 2017). Manabe utiliza esta técnica para reconstruir imágenes que llegan a su mente mientras escucha música y otras actividades contemplativas. No nos detendremos en el estudio de esta obra ya que esto se escapa del objetivo principal de este artículo. No obstante, consideramos que ayuda a observar el espacio que el arte computacional ocupa en la actualidad.

Por otro lado, el compositor y artista Ryoji Ikeda está considerado como uno de los representantes internacionales contemporáneos más importantes en lo que se refiere al arte de los nuevos medios y la creación de música electrónica experimental. Los principios de su trabajo como artista se caracterizan por la realización de instalaciones inmersivas en las que envuelve al espectador en proyecciones de imagen en movimiento y elementos escultóricos y arquitectónicos que desbordan la escala humana. Esto es combinado con una producción sonora que invita a pensar en patrones matemáticos meticulosamente elaborados. Se trata de un autor implicado en un proceso de investigación que explora las posibilidades de las nuevas tecnologías digitales sometiéndolas a un análisis científico. En su proceso creativo encontramos una fuerte implicación artística centrada en la “exploración de los límites de la percepción humana y en el análisis de la naturaleza y las formas fundamentales del sonido” (Vertedor-Romero y Alonso-Calero 2021, 324).

Ryoji Ikeda es uno de los principales referentes artísticos implicados en la hibridación del arte y la ciencia a través del campo de la computación. Un claro ejemplo de esto podemos encontrarlo en su proyecto, *supersymmetry*, del año 2014. Se trata de una obra compuesta por una serie de trabajos concebidos como instalaciones audiovisuales y que pueden entenderse como la evolución de su *Live Performance, superposition 2012*. Este proyecto puede

considerarse una actualización del proceso del trabajo realizado durante una estancia de residencia de Ikeda durante el periodo 2014–15 en el instituto CERN, donde recibió el Prix Ars Electronica Collide @ Cern 2014 (Ikeda 2014).

Preguntas de investigación

Este marco contextual nos invita a plantearnos preguntas con respecto al paradigma del arte computacional en la actualidad. Entendemos que el trabajo colaborativo resulta imprescindible en diferentes prácticas artísticas, esto se pone aún más de manifiesto en obras en las que se tiene en cuenta la computación como recurso plástico. En este tipo de obras, generalmente se requieren equipos interdisciplinarios en los que el trabajo colaborativo suele llevarse a cabo entre artistas e ingenieros informáticos y precisa de unos códigos de lenguaje que a menudo resultan ajenos al ámbito artístico. Por este motivo, una de las principales cuestiones que se desprenden de nuestro recorrido en la creación artística en este contexto computacional es, ¿debe considerarse necesario introducir el lenguaje de programación visual en el ámbito de las Bellas Artes? Nuestra principal respuesta a esta pregunta es afirmativa, ya que en general encontramos una carencia en este sentido en este entorno. Consideramos esencial comprender este tipo de lenguajes en el caso de artistas que decidan introducirse en este tipo de obra, esto facilitaría incluso la comunicación entre diferentes disciplinas como ya se ha comentado. Este trabajo pretende construir un argumento para apoyar nuestra postura en cuanto a esta afirmación que sostenemos como respuesta.

De este contexto se desprende otra pregunta que consideramos bastante pertinente, esto es, ¿se requieren nuevas metodologías docentes en las que el trabajo con código y nuevos medios sirvan para generar obra en la que la tecnología no se manifieste como un elemento que sobrepase a la propia obra? Entendemos que no podemos dar una respuesta a esta pregunta debido a que esto requiere un trabajo dedicado a este tema, no obstante señalaremos algunas líneas de investigación que se encuentran trabajando en esta dirección a través del estudio de nuevas metodologías aplicadas a este fin. Se asume así la necesidad de realizar una investigación en este sentido ya que el presente trabajo abre las vías para esto.

Objetivos y Justificación

El objetivo principal de este trabajo es establecer un contexto creativo vinculado a la tecnología y al ámbito de la computación creativa a través de las posibilidades multidisciplinares del lenguaje de programación visual (VPL en adelante por sus siglas en inglés, *Visual Programming Language*) en general y de la herramienta, VVVV, *a multipurpose toolkit* en particular. Su aplicación en entornos creativos vinculados a las enseñanzas del campo del arte o la comunicación visual es sin duda una aportación imprescindible para artistas o usuarios cuyas inquietudes deriven hacia el terreno de los nuevos medios computacionales y se sientan atraídos por las posibilidades creativas que ofrece el código de programación.

Otro objetivo de este trabajo es poner de manifiesto la importancia que tiene el aprendizaje de diferentes lenguajes de programación visual para creadores y artistas multimedia contemporáneos en el ámbito artístico. Uno de los retos pedagógicos a poner en práctica con este perfil de estudiantes sería mantenerlos motivados tras la fase inicial de aprendizaje de los conceptos básicos y en la creación de los primeros programas sencillos con VVVV. Esto se debe a que, normalmente, este perfil de alumnado ya posee sus propios medios de expresión artística y experimentan un desajuste entre su habilidad actual y lo que necesitarían para aplicar con éxito la programación audiovisual en su producción artística.

Partimos de la hipótesis de que el código de programación es imprescindible para cualquier creador audiovisual contemporáneo y consideramos la herramienta, VVVV, un recurso

funcional y altamente operativo que permite crear este tipo de contenido, aunque se esté partiendo de un desconocimiento del código textual, pero que, por su cualidad visual, facilita al usuario la posibilidad de adentrarse en el diseño y la creación de comportamientos computacionales interactivos.

El principal motivo que nos lleva a la elección de este programa viene determinado por constituirse como una herramienta principalmente pensada para prototipar rápidamente a través de un entorno fluido. Su interfaz intuitiva permite obtener en pocos pasos resultados muy poderosos a través de los denominados *patches*. Este programa está soportado por una gran comunidad además de tener a nuestra disposición un elevado número de recursos de código libre. Al unificar el acceso a múltiples tecnologías, como pueden ser, la visión por computadora, *Machine Learning* (ML), la animación 3D, realidad virtual o realidad aumentada, VVVV las aproxima a una audiencia más amplia de personas sin experiencia previa con código de programación. Como ya hemos comentado, la característica más distintiva de las VPL es que se representan por gráficos, lo que significa que los usuarios no escriben código, sino que crean los llamados *patches*, es decir, nodos conectados por cables que en su conjunto de funciones describen un programa. Mientras que la programación tradicional con código a menudo consiste en un ciclo de editar–compilar–ejecutar, VVVV sólo conoce un modo, el tiempo de ejecución. Esto permite una manera más interactiva e intuitiva, paramétrica, de crear programas y hace que VVVV sea especialmente adecuado para aplicaciones interactivas y performativas en vivo.

Metodología

A través de una metodología de análisis de contenido, desarrollaremos un línea argumental siguiendo una dirección que apoye nuestra hipótesis de partida. Se establecerá un hilo conductor previo a modo de marco teórico en el que se mostrarán algunos recursos digitales que podemos encontrar en algunas tendencias de creación artística contemporánea vinculadas a la computación creativa. Se definirá posteriormente un contexto histórico en torno al lenguaje de programación visual estableciendo como punto de partida el lenguaje de programación Logo desarrollado en 1967 por Feurzeig, Seymour y Cynthia Solomon. Una vez establecido este contexto histórico, se procede al análisis de la herramienta poniendo el foco de atención en alguna de sus funcionalidades más creativas. Se revisarán referentes y obras artísticas en las que el uso del programa VVVV sea el recurso principal implicado en el proceso de gestión. Esto permitirá finalmente elaborar un apartado de discusiones en el que se plantea un análisis crítico del programa a partir de la observación de los puntos que serán desarrollados en este trabajo.

Recursos Digitales para la Creación Artística

Iniciamos nuestro marco teórico destacando brevemente algunos de los programas de código abierto utilizados habitualmente en proyectos de creación audiovisual. La experiencia señala la importancia de conocer diferentes herramientas ya que estas pueden ser utilizadas en función de la naturaleza de cada proyecto. Comenzaremos por destacar el clásico PD (Pure Data), utilizado en numerosas ocasiones como “herramienta para enseñar electrónica a modo de juego en combinación con microprocesadores y otro *hardware*” (Hancock 2014, 109). Cabe señalar que este programa evolucionó a su versión de pago, Max/MSP, un medio de programación gráfico desarrollado y mantenido por Cycling’74 para la “creación de música y control multimedia ampliamente utilizado por compositores, artistas y diseñadores en la creación de entornos interactivos” (Cycling’74). PD es un lenguaje de programación visual de código abierto desarrollado por Miller Puckette durante los años noventa para la creación de música interactiva y de obras multimedia por computadora. Aunque Puckette es el principal autor del programa, este proyecto “continúa implementándose en la actualidad gracias a un gran número de desarrolladores que trabajan en nuevas extensiones para este” (Pure Data).

Dentro de la tipología de *software* para la creación audiovisual, existe también un amplio abanico de programas cuyo código está basado en texto que nos gustaría al menos señalar en este artículo. En primer lugar, destacaremos el programa Supercollider. Se trata de un lenguaje de programación altamente personalizable empleado para la creación de música electrónica en prácticas de Live Coding por su capacidad de reestructuración dinámica del código en directo, lo que ofrece múltiples posibilidades musicales. “Este programa ha sido utilizado también para la enseñanza de música computacional por diferentes autores” (Collins 2016, 13). Por otro lado, encontramos el programa TidalCycles, desarrollado por el músico, investigador y creador de la Algorave, Alex McLean. Se trata de un lenguaje específico de dominio incluido en Haskell inspirado en la *Functional Reactive Programming* (FRP). “La FRP integra el flujo de tiempo y los eventos de composición en la programación funcional que proporciona una manera elegante de expresar la computación en diferentes campos, como la animación interactiva, la robótica, la visión por computadora, las interfaces de usuario o la simulación” (Blackheath y Jones).

Haskell es un lenguaje de programación informática específicamente diseñado para manejar una amplia gama de aplicaciones, desde numéricas hasta simbólicas. “Haskell cuenta con una sintaxis expresiva y una rica variedad de tipos de datos incorporados” (Haskell). En TidalCycles el tiempo es racional, de manera que las subdivisiones musicales pueden ser almacenadas con precisión como fracciones simples. Detrás de esto subyace la suposición de que el tiempo está estructurado en términos de ciclos rítmicos o más correctamente métricos, un fenómeno perceptivo que yace en la base de muchas tradiciones musicales, abarcando desde las clásicas estructuras musicales indias hasta las más actuales producciones de música electrónica. “El primer tiempo de cada ciclo, conocido como *sam*, es significativo tanto para resolver el ciclo anterior como para comenzar el siguiente. La línea numérica representada por enteros muestra ritmos sucesivos de *sam*” (McLean 2014, 64).

Por último, nos gustaría resaltar el programa Sonic Pi, un lenguaje simple desarrollado inicialmente para la microcomputadora Raspberry Pi que ha sido utilizado en numerosos proyectos de formación musical. El objetivo del proyecto Sonic Pi era crear un lenguaje específico mínimo (DSL [*Domain-Specific Languages*]) que proporcionara una experiencia musicalmente atractiva para los estudiantes, además de mostrar una gama razonablemente amplia de conceptos computacionales básicos. Durante su desarrollo se probó en una clase con niños de doce años que no habían tenido previamente ninguna experiencia con programación ni computación técnica. El objetivo principal era llegar a crear un programa operativo que pudiese generar una pieza musical satisfactoria. Todos los elementos sintácticos y semánticos del lenguaje fueron diseñados con estos objetivos principales. “El DSL resultante tiene algunos principios funcionales, pero también algunas características imperativas con las que se cumplen objetivos educativos específicos” (Aaron y Blackwell 2013, 36).

Software para la creación artística audiovisual

Podemos afirmar que “el *software* ha facilitado la creación, la publicación, el acceso, la distribución y la remezcla de diferentes tipos de medios como pueden ser imágenes, secuencias de imágenes en movimiento, formas en 3D, mapas o elementos interactivos” (Manovich 2013). En cuanto al desarrollo de *software* para la producción audiovisual, encontramos un amplio abanico de recursos en el mercado con características propias, esto hace que sea imprescindible la familiarización con los términos que cada programa utiliza para poder elegir la herramienta adecuada que mejor se adapte a nuestras necesidades. Dentro del terreno del *software* libre encontramos numerosas opciones que nos permiten configurar nuestros propios recursos, como ocurre por ejemplo con el ya mencionado Sonic Pi.

En el terreno de la producción de música electrónica, por ejemplo, existe un amplio repertorio de *software* DAW (Digital Audio Workstation) comercializado. Algunas empresas

punteras dedicadas al desarrollo de este tipo de *software* en la actualidad son, Ableton, Native Instruments, Steinberg, Image Line o Propellerhead. Cada una de estas herramientas proporcionan recursos virtuales y físicos que extienden los formatos de creación audiovisual. Si observamos la evolución del *software* de audio desde sus inicios hasta nuestros días se pone de manifiesto la “existencia de tecnologías con las que hoy podemos crear nuestro estudio de música basado íntegramente en software” (MusicRadar 2018). La pertinencia de toda esta tipología de programas es que permiten el trabajo en entornos híbridos pudiendo vincular programas a través de diferentes protocolos de comunicación. La siguiente imagen representa un pequeño espectro de este entorno al que hacemos referencia (figura 1):

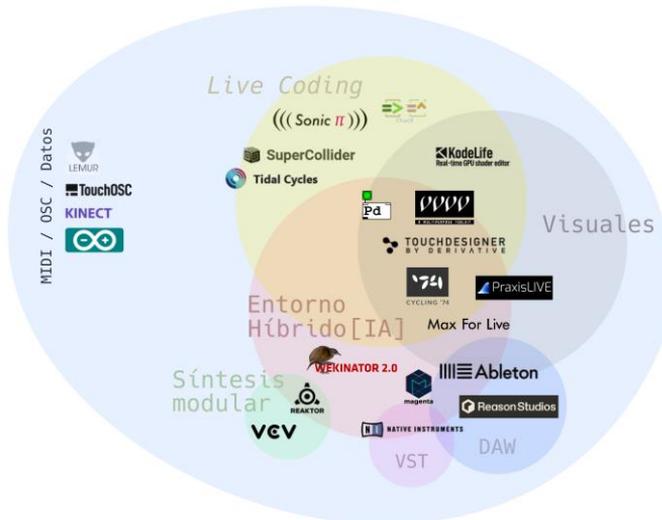


Figura 1: Diagrama de Euler en el que mostramos diferente software desarrollado para el trabajo creativo organizado por su tipología de uso en un entorno híbrido de creación
 Fuente: Elaboración propia, 2022.

Contexto histórico de los Lenguajes de Programación Visual (VPL)

Antes de pasar al estudio de la herramienta VVVV, se considera conveniente definir en qué consiste el VPL. Aunque en el punto anterior se han mostrado algunos casos de este lenguaje de programación, como es el caso de PD o Max, es necesario atender a una definición concreta de este lenguaje antes de profundizar en la herramienta que se plantea analizar en esta propuesta. Un VPL es cualquier lenguaje de programación que permite la manipulación del código subyacente de una manera gráfica en lugar del enfoque tradicional basado en texto (Noone, M., y Mooney, A. 2017, 158). Muchos de los VPL que podemos encontrar en la actualidad están basados en la idea de conectar cajas a través de líneas que representan relaciones entre estas cajas o módulos. Este modelo de lenguaje de programación se produce en paralelo a la aparición de la informática y su aplicación en el arte. Su objetivo ha sido el proceso de facilitar el uso de los lenguajes de programación a todas las personas. Algunas de estas iniciativas se han centrado explícitamente en rebajar el umbral de aprendizaje y a fomentar un enfoque diferente de la codificación. Podría afirmarse que este método de programación se parece mucho al enfoque que ofrece el bricolaje.

En los orígenes del desarrollo de las VPL, la intención era despertar la pasión por la programación en los niños desde una edad temprana. Por este motivo, Feurzeig, Seymour y Cynthia Solomon diseñaron el lenguaje de programación Logo en 1967 (Solomon y Papert 1976). Logo es famoso por el uso que hace de lo que se conoce como gráfica tortuga, en la que

“los comandos de movimiento y dibujo producían gráficos de líneas o vectores en una pantalla” (Daskalov, R., Pashev, G., y Gaftandzhieva, S. 2021, 981). Más tarde, aparecieron las VPL basadas en bloques, como es el caso del programa Scratch. Además de ser una VPL basada en bloques, Scratch ofrece un sitio web dirigido principalmente a niños de 8 a 16 años como herramienta educativa para la codificación. Esta es una comunidad en línea diseñada principalmente para niños, donde los usuarios crean proyectos en línea utilizando una interfaz similar a la de los bloques. Scratch permite a los profesores organizar los proyectos de los alumnos en estudios. Las estadísticas de la comunidad en el sitio web oficial de Scratch muestran que más de 50 millones de proyectos de Scratch son compartidos por más de 50 millones de usuarios. Scratch proporciona funciones adicionales como la voz y el movimiento del usuario con kits de invención específicos para llevar a cabo la interacción ciberfísica. Existen proyectos como el caso de EduTalk que utilizan estas características de interacción ciberfísica para integrarse con Scratch a través de un bloque de extensión (Lin et al. 2022).

VVVV, una herramienta híbrida multipropósito

VVVV es un sistema de programación visual centrado en la generación de gráficos y animaciones, así como en la creación de interfaces que faciliten la comunicación entre el humano y la máquina. Se trata de un programa desarrollado por Meso, una firma alemana que, como ellos mismos definen, está formada por un equipo de exploradores creativos, diseñadores, estrategas y creadores listos para ayudar a dar forma al futuro de las tecnologías digitales. Fundada en 1997 por “diseñadores que aman las computadoras y por científicos de la computación con una debilidad por el diseño, el objetivo de esta compañía es unir disciplinas y crear *hardware* y *software* para cosas y espacios comunicativos” (MESO).

Este entorno de programación está basado en la visualización de flujo de datos con un enfoque de trabajo en vivo, es decir, el programa ya está compilado durante la construcción y su ejecución, lo que produce que cada cambio afecte inmediatamente en el resultado. Esto lo hace particularmente interesante para el uso creativo, ya que “continuamente proporciona retroalimentación visual y, por lo tanto, el programa puede modificarse en directo en función del resultado” (Jung y Oschatz 2007). Esta funcionalidad hace que suela utilizarse también en diseño paramétrico de geometrías. Otra característica interesante de este programa es su cualidad híbrida, esto significa que a un nivel inferior del código visual del programa se alojan dos códigos de programación textual (figura 2) con los que se pueden conseguir resultados más complejos trabajando a un nivel más bajo con la máquina, estos son: High-Level Shading Language (HLSL) y C Sharp (C#).

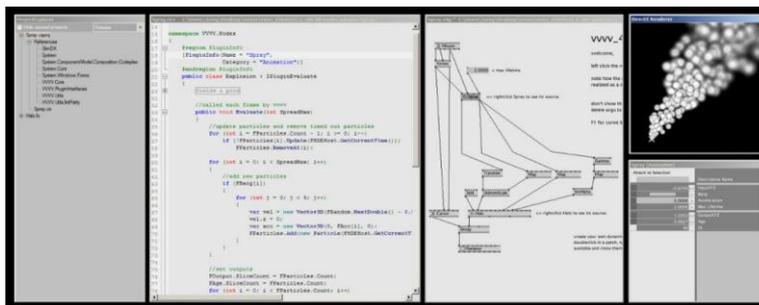


Figura 2: De izquierda a derecha encontramos el explorador de proyectos, continuación el código C#, seguido del código modular, que se encuentra en un nivel superior al textual. Por último, vemos un renderer, éste permite visualizar el resultado gráfico del código. Debajo de esta pantalla encontramos el *Herr Inspektor*. A través de él podemos manipular los parámetros que están programados en el nodo.

Fuente: Recuperado de (15 de junio, 2019) <https://vvvv.org/screenshots>, 2019.

VVVV se utiliza en áreas diferentes, como pueden ser: visión por computadora, visualización de datos, computación física, animación 2d / 3d en tiempo real, configuraciones multipantalla, realidad virtual, realidad aumentada o ML. Tecnológicamente el programa se basa en .NET y por tanto permite la interacción directa con todo tipo de bibliotecas disponibles para esta plataforma de aplicaciones. Por mencionar solo algunos de los recursos que permite conectar destacaremos los siguientes: Xenko (motor 3d), OpenCV (visión por computadora), Bullet-Physics (física 3d), Box2d (física 2d), OSC (Open Sound Control), Art-Net, Midi, Arduino / Firmata (Starts).

Desde la web del programa VVVV, afirman que se trata de una de las formas más adecuadas de poner en contacto a las personas con temas como la codificación creativa, el diseño generativo, los gráficos por computadora, el diseño de interacción, la visualización de datos, la visión por computadora, la computación física, el aprendizaje de máquina y similares. Entre las ventajas que se señalan, cabe destacar que se trata de un programa gratuito para uso no comercial o educativo para el que no hace falta registro; se puede ampliar fácilmente con nodos personalizados escribiendo código C # estándar o utilizando casi cualquier *nuget* de .NET; todas sus bibliotecas son de código abierto, lo que significa que se pueden explorar, aprender y ampliar; utiliza conceptos de programación estándar de la industria conocidos de la programación orientada a objetos que, una vez comprendidos, también se pueden aplicar en otros lenguajes de programación; los desarrolladores principales y la excelente comunidad de usuarios “ofrecen soporte de ayuda directa a través del chat y el foro básicamente 24/7/365” (Joreg 2020). VVVV se está utilizando en algunos entornos educativos para aprender conceptos de programación, resolución de problemas y pensamiento con computadoras. Otros usos que se hace de este programa en este ámbito es el procesamiento gráfico generativo y en tiempo real, visualización de datos, instalaciones, diseños sonoros reactivos, *Motion graphics*, procesamiento de vídeo en tiempo real o realidad aumentada, entre otras técnicas (VVVV).

Espacios de intercambio de conocimiento

El mundo del *software* está amparado por numerosas comunidades ubicadas en los foros que ofrecen los sitios web de los creadores de cada programa. Este formato resulta ser un extraordinario método de aprendizaje e intercambio de conocimientos, experiencias y resultados obtenidos en experimentos o proyectos realizados por diferentes usuarios. De esta manera, estos programas se convierten en recursos al alcance de cualquier persona y fomentan el autoaprendizaje del código de programación y de nuevos procesos creativos computacionales. Podemos ver un ejemplo claro en la red de socios internacionales, NODE, *Verein zur Förderung Digitaler Kultur e.V.*, una organización fundada en 2010 que se registró sin fines de lucro cuya sede se encuentra Frankfurt. Desde su aparición el consejo está en manos de David Brüll y Sebastian Oschatz.

La junta de organización de NODE es voluntaria y tanto los numerosos miembros honorarios del equipo que componen esta asociación como sus anfitriones creen en el espíritu de compartir conocimientos e intercambiar ideas. Sus intereses se centran en la creación de oportunidades para un intercambio abierto de las investigaciones actuales basadas en el encuentro entre cultura, arte y tecnología. Además, desarrollan formatos que se centran en la educación con la finalidad de ofrecer métodos prácticos de codificación, abordando temas sociopolíticos y discutiendo sobre nuevos desafíos para las prácticas artísticas y de diseño que acompañan a la transformación digital de nuestra sociedad (NODE).

Cabe mencionar que NODE ofrece oportunidades de residencias dentro de una red de festivales patrocinadas por sus socios internacionales o con el apoyo del *Goethe Institut*. A través de formatos como las sesiones NODE + CODE, se fomenta el continuo conocimiento relacionado con prácticas digitales y de medios de comunicación en Frankfurt y la región de

Rhine–Main. Además, esta asociación cuenta con la colaboración de universidades locales y organizaciones culturales a través de las cuales se invita a artistas inspiradores para que puedan compartir sus conocimientos y desafíos con artistas locales, estudiantes y el público interesado (NODE, s.f.). A lo largo de su historia, NODE se ha convertido en un foro imprescindible como punto de encuentro entre profesionales, estudiantes, programadores, creativos, artistas y activistas de todo el mundo. Además de ser la base del apoyo continuo de la comunidad de expertos en programación con VVVV, ofrecen talleres para estudiar lenguajes de programación visual y diversos métodos de codificación creativa.

Ejemplo de estructura de un sistema creativo implementado con VVVV

Para mostrar un ejemplo del flujo de trabajo con VVVV tomaremos la tesis titulada, *Entwurf und Implementierung einer zeitleistenbasierten Parametersteuerung in eine datenstromorientierte Echtzeitprogrammierungsumgebung*⁵, del profesor de diseño computacional y computación física, Ingolf Heinsch (MILO). En este trabajo, el autor investiga la posibilidad de combinar un entorno de programación en tiempo real orientado al flujo de datos con el concepto de control de parámetros basado en la línea de tiempo. Básicamente, el sistema consta de tres niveles diferentes (figura 3). El nivel de control, que integra los componentes de la lógica de reproducción y funciona de forma autónoma al resto del sistema. El nivel de gestión se especifica como el componente principal de la aplicación, en este nivel, toda la comunicación se organiza con VVVV que se encarga de crear los pines de entrada y salida que forman la base de la interfaz. La gestión de datos también forma parte del plano de gestión. Por último, el plano de datos se encarga de manejar las trazas de datos, los fotogramas clave y sus interpolaciones (Heinsch 2007).

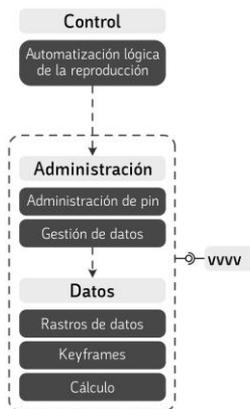


Figura 3: Arquitectura del sistema planteada por Ingolf Heinsch (Heinsch 2007, 59)

Fuente: Imagen de elaboración propia a partir de la realizada por el autor de la tesis mencionada, 2007.

Ámbitos de aplicación

Como ocurre con otros VPL para la creación audiovisual como pueden ser Max o TouchDesigner, la posibilidad de generar diferentes proyectos según las necesidades de cada artista es uno de los principios que hacen de estas herramientas un elemento fundamental para el

⁵ Traducción de los autores: Diseño e implementación de una parametrización basada en la línea de tiempo en un entorno de tiempo real orientado al flujo de datos.

proceso creativo de cualquier productor audiovisual contemporáneo. Resulta imposible abarcar en este artículo todas las posibilidades que la herramienta VVVV ofrece a este perfil de usuario, por lo que destacaremos en este apartado algunas de las funciones del programa que pueden ser útiles para el desarrollo de ciertos proyectos creativos a diferentes escalas.

En primer lugar, analizaremos la arquitectura cliente–servidor que incorpora VVVV llamada, *boygrouping*. Este sistema permite controlar cualquier número de computadoras en procesamiento (clientes) desde un solo servidor (figura 4). El flujo de trabajo con *boygrouping* funciona de manera que los *patches* se programan solo en el servidor, mientras que VVVV se encarga de todos los clientes conectados. Las aplicaciones de *boygrouping* suelen ser distribuciones de múltiples pantallas o proyecciones. En configuraciones avanzadas, puede ser necesario que un cliente escuche múltiples servidores, lo que se puede lograr usando otro sistema que no abordaremos en este trabajo, *MultiBoygrouping*.

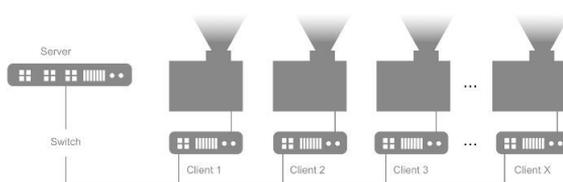


Figura 4: Esquema de uso de la arquitectura boygrouping con comunicación servidor / client
Fuente: Imagen propia, 2022.

En lo que se refiere al procesamiento de sonido, VVVV no es una herramienta especialmente dedicada a esta función. Para este propósito existen programas como PD o Max, por ejemplo, ya que están desarrollados para trabajar con más precisión en la tarjeta de sonido. Los nodos de audio nativos de VVVV son de la categoría *Dshow9*, basados en *DirectShow*, una arquitectura de *streaming* para Microsoft Windows que utilizan las aplicaciones para poder realizar una reproducción o captura de audio y vídeo de alta calidad. Pueden encontrarse motores de audio en diferentes paquetes. Por ejemplo, “VVVV.Audio Pack Alpha” ofrece opciones de enrutamiento flexibles y transparentes junto con un sólido soporte para controladores VST y ASIO. Otros motores de audio que podemos descargar y utilizar dentro del programa son por ejemplo la biblioteca, Bass y la biblioteca de audio multiplataforma de sonido 2D/3D de alto nivel, irrKlang.

El programa también ofrece herramientas muy completas para analizar el audio de entrada. Encontramos analizadores de envolvente simples que toman el valor de la potencia RMS⁶ (Root Mean Square) del sonido, como son los nodos *Beat trackers*, que pueden ser útiles para rastrear ritmos o los nodos FFT. Una FFT devuelve una extensión de amplitudes de una escala de frecuencias distribuida linealmente. Como Podemos ver en la siguiente imagen (figura 5), el analizador gráfico puede servirnos para observar las frecuencias que están siendo activadas y así programar comportamientos basados en estos datos para nuestro proyecto.

⁶ Se denomina así a la potencia real del sonido. Nota de los autores.

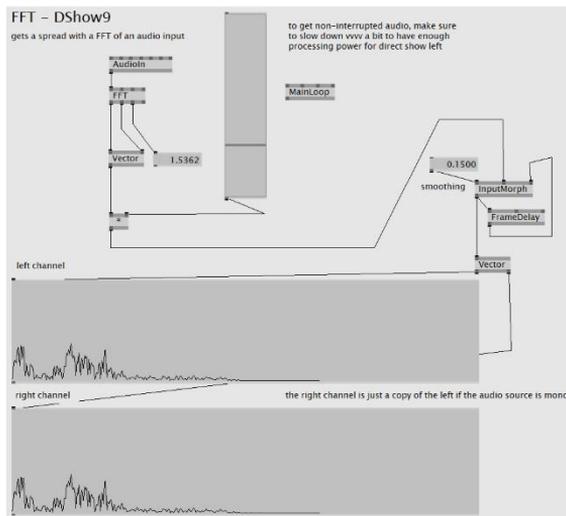


Figura 5: Captura de pantalla tomada del analizador FFT – Dshow9 ejecutándose en el programa, 2022.

VVVV en la creación artística multi- e interdisciplinar

Nos gustaría concluir este artículo destacando algunos proyectos realizados con el programa VVVV. Con esto pretendemos resaltar la enorme versatilidad de esta herramienta digital además de mostrar un flujo de trabajo con numerosas opciones creativas. En primer lugar, prestaremos atención al artista programador Sebastian Neitsch, nacido en Hamburgo en 1982. Estudió Diseño de interfaces multimedia en Halle y Linz. Desde 2008 trabaja en proyectos basados entre el diseño y la tecnología, tratando de encontrar nuevos conceptos para las interacciones hombre-máquina. Ha trabajado como profesor en la Universidad de Arte de Linz y en la Universidad Bilgy en Estambul. De este autor señalaremos su proyecto Episireo, realizado para el Ars Electronica 2008 en Linz en el Holmes Place Health Club. En este trabajo, Neitsch trabajó en colaboración con los artistas audiovisuales, Stefan Schwabe y Fabien Artal. Las herramientas utilizadas en esta ocasión fueron VVVV para los gráficos y Pure Data para el sonido. El concepto básico de la instalación era el de crear gráficos digitales generados por el movimiento de los nadadores. Además, desarrollaron diferentes modos y comportamientos complejos dependiendo del tiempo y la intensidad del movimiento del nadador. Sobre la superficie y debajo del agua se colocaron altavoces que reproducían sonidos de audio generados en vivo (Neitsch).

En la línea de investigación relacionada con la sonificación y visualización de flujo de datos, nos gustaría destacar también al autor Roberto Vitalini, del estudio, Bashiba, dedicado a la creación e integración de gráficos en movimiento en tiempo real en los campos del teatro, la ópera y otros eventos en directo. En su proyecto, Bashiba Neuroheadset 2016, Vitalini colabora con el artista Sebastiano Barbieri para construir un entorno audiovisual basado en la visualización de datos de EEG (electroencefalograma) en tiempo real y la sonificación de estos datos (Bashiba 2016). A través del protocolo de comunicación de código libre OSC, VVVV permite gestionar los datos de entrada. De esta manera es posible diseñar los diferentes comportamientos que pueden aplicarse a los datos para configurar finalmente el entorno audiovisual que se obtiene como resultado. Una de las opciones de *hardware* disponibles para desarrollar este entorno con el código de programación de VVVV, se hace viable mediante la conexión del sensor Emotiv EPOC, diseñado para la investigación del cerebro humano. Este sensor, “proporciona acceso a datos cerebrales de una manera sencilla y profesional” (Emotiv).

En la página web de VVVV encontramos el código para ejecutar esta operación y obtener los datos del sensor Emotiv EPOC en VVVV a través del protocolo de comunicación OSC (lanvideosource 2013).

Uno de los artistas destacables en este contexto es el autor turco de los nuevos medios, Refik Anadol, quien trabaja con diferentes sistemas de inteligencia artificial que programa en numerosas ocasiones con la herramienta VVVV. Su estudio artístico es conocido por transformar espacios arquitectónicos en lienzos gigantes. El corpus conceptual de su trabajo aborda los desafíos y las posibilidades de la computación y el significado del ser humano en la era de la inteligencia artificial. Uno de sus trabajos ejemplares es *Melting Memories*, una serie de obras de arte digitales con las que el artista explora la materialidad del recuerdo. En estas esculturas de datos aumentados y proyecciones de luz, Anadol ofrece al espectador la posibilidad de experimentar interpretaciones estéticas de datos de EEG recopilados en los mecanismos neuronales de control cognitivo (AIArtists). Cabe señalar que algunas de sus instalaciones las realiza a través de los servicios ofertados por el estudio alemán, Analog Native.

Entre las opciones que encontramos disponibles para este programa, hallamos que, en su versión gamma, VVVV cuenta con la posibilidad de instalar aplicaciones para conectar con programas que trabajan con ML. Un ejemplo de esto son los nodos para conectar con el programa Wekinator 2.0, un meta instrumento desarrollado por la profesora del Instituto de Computación Creativa de la Universidad de Artes de Londres, Rebecca Fiebrink. Esta herramienta permite que “cualquier persona utilice ML para construir nuevos instrumentos musicales, controladores de juegos gestuales o sistemas de visión o escucha por computadora, entre otras aplicaciones” (Fiebrink 2019). Existen numerosos trabajos en los que se puede observar el uso de esta herramienta en su versión más actual que nos permite ver sus posibilidades interdisciplinarias a través de metodologías computacionales en la elaboración de una obra artística. Un ejemplo de esto es el proyecto desarrollado por Emanuel Gollob, Magdalena Mayer y Johannes Braumann, en el que encontramos un ejemplo de “control robótico generativo (KUKA|prc), una Red Generativa Adversarial Convolutiva Profunda y la electroencefalografía (EEG) para crear una estrategia de adaptación estética dentro de un espacio de salida paramétrico físico” (Gollob y Braumann 2021)

Discusión

Como se ha mencionado en este artículo, existen numerosas herramientas dedicadas a la creación artística que ya han sido numeradas a lo largo de este texto. Consideramos que VVVV en su versión Beta facilita el acceso a cualquier persona sin conocimientos previos en lenguajes de programación abriendo un terreno a la creación artística computacional que servirá de punto inicial para emprender tanto proyectos en solitario, como el caso de VertexZenit en su proyecto [DIYSIK], donde se aborda un proyecto de creación artística basado en código abierto (Vertedor–Romero 2019), como proyectos más complejos en los que se requiere de un equipo interdisciplinario que permite abordar la complejidad planteada por el proyecto, como el caso del ya mencionado estudio de Refik Anadol.

Encontramos en este tema principios que señalan un campo de estudio en el que se necesitan explorar nuevas metodologías docentes para la aplicación de la tecnología en el ámbito de las Bellas Artes. Un ejemplo de esto lo encontramos en el grupo de investigación, Post Internet Arts Education research, iniciado por Torsten Meyer en la Universidad de Colonia, desde el que apuestan por educación artística digital y post–digital desde 2015, constituyendo de este modo una importante contribución a la investigación básica sobre el arte post–internet y la cultura post–digital (Meyer 2021). Como afirma la profesora de la Universidad de Colonia, Alemania, Kristin Klein, “las obras de arte que reflejan las culturas digitales permiten otras formas de imaginar o tratar la digitalización” (Klein 2021, 34). Entendemos que este estudio

requiere ser abordado desde una propuesta dedicada a este tema por la extensión que necesita este campo de estudio.

Conclusiones

Con este trabajo hemos tratado de poner el énfasis en señalar algunas de las nuevas posibilidades que la tecnología actual nos proporciona, en el sentido de abrir un mayor campo de interacción y aportar novedosos recursos creativos que pueden ser aplicados en diferentes campos del pensamiento desde la producción audiovisual. Además, se demuestra que, en el terreno audiovisual, el *software* libre e Internet nos han brindado nuevas formas de comprender y producir contenido audiovisual, editarlo y distribuirlo por la red. De esta manera se pone de manifiesto la necesidad de profundizar en el conocimiento de nuevas herramientas digitales con la finalidad de alcanzar un uso óptimo de éstas y obtener resultados interesantes y de gran impacto artístico. Es un hecho que se hace cada vez más presente la necesidad depositada en el ser humano de conocer tecnologías habilitadoras que simplifiquen la actividad digital en un entorno social cada vez más dirigido por la tecnología.

Como hemos podido ver a lo largo de este trabajo, existen numerosas herramientas de código libre que permiten un flujo de trabajo en diferentes disciplinas creativas, no obstante, VVVV se presenta como un instrumento enfocado a artistas o creadores audiovisuales que se están iniciando en los lenguajes de programación o que no han tenido nunca contacto con ellos, además de ofrecer recursos más complejos a usuarios avanzados gracias al código de programación C#, como hemos podido observar. El análisis de esta herramienta demuestra su versatilidad para la aplicación en diferentes propuestas creativas, por lo que sostenemos la idea de que se trata de un programa que puede ayudar de manera muy positiva a artistas audiovisuales en el desarrollo de proyectos de complejidad en distintos niveles.

Gracias a la gran comunidad que apoya este programa, resulta muy natural involucrarse en un proceso creativo digital en cualquier línea de investigación. Debido a que utiliza librerías de *software* libre, podemos encontrar numerosos recursos o prototipos que otros usuarios han desarrollado y liberado para que puedan ser utilizados, implementados o modificados por otros productores. Este hecho, añadido a la rápida respuesta de cualquier miembro de la comunidad ofrece ante cualquier duda, permite utilizar los *patches* de una manera creativa y habilita para la creación de nuestro prototipo o programa.

Por todo lo visto en el desarrollo de este artículo, consideramos VVVV una herramienta muy completa y con un gran potencial para la creación computacional que facilita su uso a artistas audiovisuales. El manejo de este programa resulta intuitivo y pueden obtenerse resultados atrayentes utilizando tan sólo algunos recursos. Algo importante que debemos señalar es que, como ocurre con el aprendizaje de cualquier código de programación o de cualquier disciplina, se necesita un periodo de prácticas con éste para alcanzar los niveles de complejidad deseados, sin embargo, consideramos que la curva de aprendizaje con VVVV puede incrementarse muy rápido en poco tiempo. Por otro lado, sostenemos la idea de que, en el aprendizaje de este tipo de herramientas digitales, se acentúan los procesos de autoaprendizaje gracias a los recursos ofrecidos por las comunidades de usuarios que ya hemos citado.

Consideramos que este estudio confirma la primera pregunta de investigación que planteábamos al principio. Encontramos una necesidad de introducir terminología utilizada en materia informática resulta un material que debe ser introducido en las enseñanzas artísticas a la vez que se implementa dentro de este marco el lenguaje de programación visual. Del mismo modo, podemos afirmar que las múltiples aportaciones que el uso de la tecnología traen a este territorio requieren de nuevas metodologías docentes, como se ha señalado con los trabajos de Meyer y Klein. Esto abre la investigación a explorar estas metodologías e incluso a la elaboración de metodologías híbridas que podrán constituir un nuevo campo de estudio.

Agradecimientos

Esta investigación se ha desarrollado en el marco de la Ayuda de Formación de Jóvenes Doctores Margarita Salas, concedida por la Universidad de Málaga y el Ministerio de Universidades de España, durante una estancia de investigación en la Facultad de Bellas Artes de la Universidad de Granada.

REFERENCIAS

- Aaron, S., y F. A. Blackwell. 2013. “From Sonic Pi to Overtone: Creative Musical Experiences with Domain-specific and Functional Languages”. En *Proceedings of the First ACM SIGPLAN workshop on Functional art, music, modelling & design – FARM ’13*, 35–46. <https://doi.org/10.1145/2505341.2505346>
- AIArtists. Refik Anadol – Artist Profile | AiArtists. Recuperado el 22 de marzo de 2022, de <https://aiartists.org/refik-anadol>
- Ardoy, G. No Digas Artista Digital, di Daito Manabe. Recuperado el 31 de julio de 2019, de <http://www.byfanzine.com/daito-manabe/>
- Atrium, Z. K. M. 2015. June 21, 2015–August 9, 2015, Ryoji Ikeda. micro | macro.
- Bashiba. 2016. Real-Time EEG Data Visualization + Sonification | vvvv. <https://vvvv.org/blog/real-time-eeg-data-visualization-sonification>
- Benjamin, W. 1982. *La Obra de Arte en la época de su Reproductibilidad Técnica*. Discursos interrumpidos, Taurus Ediciones.
- Blackheath, S., y A. Jones. S. f. Functional Reactive Programming. Recuperado el 4 de octubre de 2019, de https://wiki.haskell.org/Functional_Reactive_Programming
- CERN. S. f. Supersymmetry | CERN. Recuperado 20 de abril de 2018, de <https://home.cern/about/physics/supersymmetry>
- Collins, N. 2016. “Live Coding and Teaching SuperCollider”. *Journal of music, technology, and education*, 9(1), 5–16. <https://doi.org/http://dx.doi.org/10.1037/xge0000076>
- Compart. Oscillon No. 40 | Database of Digital Art. Recuperado el 17 de abril de 2021, de <http://dada.compart-bremen.de/item/artwork/533>
- Cycling’74. Cycling ’74. Recuperado el 4 de octubre de 2017, de <https://cycling74.com/>
- Daskalov, R.; G. Pashev, y S. Gaftandzhieva. 2021. “Hybrid Visual Programming Language Environment for Programming Training”. *TEM Journal*, 10(2): 981–986. <https://doi.org/10.18421/TEM102-63>
- Emotiv. EMOTIV EPOC+ 14–Channel Wireless EEG Headset | EMOTIV. Recuperado el 17 de mayo de 2021, de <https://www.emotiv.com/epoc/>
- Fiebrink, R. 2019. “Machine Learning Education for Artists, Musicians, and Other Creative Practitioners”. *ACM Transactions on Computing Education*, 19(4). <https://doi.org/10.1145/3294008>
- Gollob, E., M. Mayer, y J. Braumann. 2021. “Using Robotics and A.I. to Physically Explore a Space of Aesthetic Possibilities: Defining a Physical Aesthetic Experience by the Targeted EEG Feedback of the Perceiver. En *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI ’21)*. Association for Computing Machinery, New York, NY, USA, Article 28, 1–8. <https://doi.org/10.1145/3430524.3440647>
- Gombrich, E. H. 2008. *La Historia del Arte* (16.a ed.). Phaidon Press Limited.
- Hadjakos, A., H. Schulze, A. Düchting, C. Metzger, M. Ottensmann, F. Riechmann, A–M Schneider, y M. Trappmann. 2015. “Learning Visual Programming by Creating a

- Walkable Interactive Installation”. En *Proceedings of the Audio Mostly 2015 on Interaction with Sound – AM '15*, 1–8. <https://doi.org/10.1145/2814895.2814914>
- Hancock, O. 2014. “Learning of Pure Data: A case study”. *Journal of Music, Technology & Education Volume*, 7(1), 93–112. <https://doi.org/10.1386/jmte.7.1.93>
- Haskell. Introduction – HaskellWiki. Recuperado 4 de octubre de 2017, de <https://wiki.haskell.org/Introduction>
- Heinsch, I. 2007. “Entwurf und Implementierung einer zeitleistenbasierten Parametersteuerung in eine datenstromorientierte Echtzeitprogrammierungsumgebung”. Tesis doctoral. Fachhochschule für Technik und Wirtschaft Berlin.
- Horikawa, T., y Y. Kamitani. 2017. “Generic decoding of seen and imagined objects using hierarchical visual features”. *Nature Communications*, 8 (May): 1–15. <https://doi.org/10.1038/ncomms15037>
- Ikeda, R. 2014. Ryoji Ikeda | supersymmetry. <http://www.ryojiikeda.com/project/supersymmetry/>
- Joreg. 2020. NODE20 – Educational discounts | vvvv. Recuperado 29 de julio de 2021, de <https://vvvv.org/blog/node20-educational-discounts>.
- Klein, K. 2021. “Post-digital, Post-internet: Propositions for Art Education in the Context of Digital Cultures”. En: Tavin, K., Kolb, G., Tervo, J. (eds) *Post-Digital, Post-Internet Art and Education. Palgrave Studies in Educational Futures*. 27–44. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-030-73770-2_2
- lanvideosource. 2013. Art and Brainwaves with VVVV | vvvv. <https://vvvv.org/contribution/art-and-brainwaves-with-vvvv>
- Manabe, D. Biography | Daito Manabe. Recuperado 31 de julio de 2019, de <http://www.daito.ws/en/biography/>
- Manovich, L. 2013. *Software takes command* (1a). Bloomsbury Academic.
- McLean, A. 2014. “Making programming languages to dance to: Live Coding with Tidal”. *Proceedings of the 2nd ACM SIGPLAN international workshop on Functional art, music, modelling & design*, 63–70. <https://doi.org/10.1145/2633638.2633647>
- McLean, C. A. 2011. “Artist-Programmers and Programming Languages for the Arts”. *Control, October*. Tesis doctoral. University of London.
- MESO. MESO – About us. Recuperado el 23 de marzo de 2019, de <https://meso.design/en/pages/about-us>
- Meyer, T. 2021. “A New Sujet/Subject for Art Education”. En: Tavin, K., Kolb, G., Tervo, J. (eds) *Post-Digital, Post-Internet Art and Education. Palgrave Studies in Educational Futures*. 131–145. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-030-73770-2_8
- MusicRadar. 2018. The best digital audio workstations for PC and Mac | MusicRadar. <https://www.musicradar.com/tuition/tech/the-20-best-daw-software-apps-in-the-world-today-238905>
- Neitsch, S. EPISUREO. Recuperado el 17 de noviembre de 2022, de <http://www.sebastianneitsch.de/index.php?/project/episureo/>
- NODE. NODE Forum for Digital Arts. Recuperado el 24 de junio de 2015, de <http://node.vvvv.org/>
- Noone, M., y A. Mooney. 2017. Visual and textual programming languages: A systematic review of the literature. *ArXiv*, 5(2), 149–174. <https://doi.org/10.1007/s40692-018-0101-5>
- Preston, S. 1965, 28 de abril. “Reputations Made and in Making”. *The New York Times*, 23. <https://timesmachine.nytimes.com/timesmachine/1965/04/18/101539366.pdf>
- PureData. Pure Data — Pd Community Site. Recuperado 4 de octubre de 2017, de <https://puredata.info/>

- Cynthia J. Solomon y Seymour Papert. 1976. A case study of a young child doing turtle graphics in LOGO. In Proceedings of the June 7–10, 1976, *National Computer Conference and Exposition (AFIPS '76)*. Association for Computing Machinery, New York, NY, USA, 1049–1056. <https://doi.org/10.1145/1499799.1499945>
- Scratch. Scratch – Acerca de. Recuperado el 23 de marzo de 2019, de <https://scratch.mit.edu/about>
- Starts. vvvv – a visual programming environment | STARTS. Recuperado el 18 de mayo de 2021, de <https://vertigo.starts.eu/calls/start-residencies-call-3/projects/vvvv-a-visual-programming-environment/detail/>
- Vertedor–Romero, J. A. 2019. Producción Audiovisual Y código Libre. [DIYSIK] De VertexZenit. *UMÁTICA. Revista Sobre Creación Y Análisis De La Imagen* 1(1):155–69. <https://doi.org/10.24310/Umatica.2018.v0i1.2965>.
- Vertedor–Romero, J. A., y J. M. Alonso–Calero. 2021. Inmersión sonora y microsonido. Estudio de caso de la obra de Alva Noto y Ryoji Ikeda. *Espacio Tiempo Y Forma. Serie VII, Historia Del Arte*, (9), 303–340. <https://doi.org/10.5944/etfvii.9.2021.30521>
- Victoria y Albert Museum. A History of Computer Art – Victoria and Albert Museum. Recuperado 17 de abril de 2021, de <http://www.vam.ac.uk/content/articles/a/computer-art-history/>
- vvvv. Analysis | documentación de vvvv beta. Recuperado 17 de mayo de 2021, de <https://betadocs.vvvv.org/topics/audio/analysis.html>
- vvvv. Educational Institutions | vvvv. Recuperado 28 de abril de 2021, de <https://vvvv.org/documentation/educational-institutions>
- Y. –B. Lin, M. –Z. Shieh, M. –F. Shih y C. –C. Cheng, "EduTalk: An IoT Environment for Learning Computer Programming and Physics," in *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21946–21957, 1 Nov.1, 2022, doi: 10.1109/JIOT.2022.3182280.

SOBRE LOS AUTORES

José Antonio Vertedor–Romero: Investigador Doctor Margarita Salas, Departamento de Arte y Arquitectura, Universidad de Málaga, Málaga, España

Isidro López–Aparicio Pérez: Profesor Titular, Departamento de Dibujo, Universidad de Granada, Granada, España