

El proceso en la producción de arte electrónico

Hugo Solís García





Casa abierta al tiempo

Universidad Autónoma Metropolitana

Rector General

Eduardo Abel Peñalosa Castro

Secretario General

José Antonio de los Reyes Heredia

Coordinador General de Difusión

Francisco Mata Rosas

Director de Publicaciones y Promoción Editorial

Bernardo Javier Ruiz López

Subdirector de Distribución y Promoción Editorial

Marco Antonio Moctezuma Zamarrón

Unidad Lerma

Rector

José Mariano García Garibay

Secretario

Darío Eduardo Guaycochea Guglielmi

*Dirección de la División de Ciencias Sociales
y Humanidades*

Mónica Francisca Benítez Dávila

*Coordinación del Consejo Editorial de la División
de Ciencias Sociales y Humanidades*

Gladys Ortiz Henderson

DOCUMENTOS

DAH

El proceso en la producción de arte electrónico

Hugo Solís García



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA
Unidad Lerma

Universidad Autónoma Metropolitana
Unidad Lerma / División de Ciencias Sociales y Humanidades

México, 2018

Primera edición, 2018

El proceso en la producción de arte electrónico

Hugo Solís García

Edición y corrección: Rebeca Ocaranza Bastida

Diseño y cuidado de la edición: Sergio Cantinca Cornejo, Fernando Cantinca Cornejo

Formación y tipografía: Sergio Cantinca Cornejo

Formación y tipografía e-Pub: José Carlos Ramírez Palacios

Diseño de portada: Sergio Cantinca Cornejo

D.R. © Universidad Autónoma Metropolitana

Prolongación Canal de Miramontes, núm. 3855

Ex Hacienda de San Juan de Dios, Delegación Tlalpan, 14387,

Ciudad de México.

Página-e: www.ler.uam.mx

D.R. © **Unidad Lerma**/División de Ciencias Sociales y Humanidades

Avenida de las Garzas, núm. 10

Col. El panteón, 52005, Lerma, Estado de México

Consejo Editorial de la División de Ciencias Sociales y Humanidades.

<cedcsh@correo.ler.uam.mx>

Esta publicación no puede ser reproducida en todo ni en parte, ni registrada o transmitida por un sistema de recuperación de información en ninguna forma y por ningún medio, sea mecánico, fotoquímico, electrónico, magnético, electroóptico, por fotocopia o cualquier otro, sin el permiso previo y por escrito de los editores.

La presente publicación pasó por un proceso de dos dictámenes (doble ciego) de pares académicos avalados por el Consejo Editorial de la División de Ciencias Sociales y Humanidades de la UAM-Lerma, que garantizan su calidad y pertinencia académica y científica.

Hecho en México/Made in Mexico

ISBN: 978-607-28-1513-1

Índice

- 15 Agradecimientos
- 17 Prólogo
- 21 Introducción
 - El libro, 25
- 1**
- 31 Arte interactivo y participación del público como agente colaborador
 - Introducción, 31 | Contexto histórico y artístico, 32 | Arte interactivo, una clasificación imprecisa, 36 | Niveles de colaboración e interactividad, 37 | Interactividad en contexto, 39 | Técnicas y recursos para la producción de arte interactivo, 44 | Conclusiones, 54
- 2**
- 59 Música y escena
 - Trío Nicrom, 59 | Improvisaciones para piano y medios electrónicos, 61 | Ensamble Crumble, 62 | Dúo Juum, 65 | Música para danza, 67
- 3**
- 73 Radiobanda
 - Descripción, 73 | Funcionamiento, 74 | Agradecimiento, 80 | Código, 80
- 4**
- 97 Ix-.-.hel
 - Descripción técnica, 97 | La transfiguración digital de la telegrafía y su imaginario, 99 | Códigos, 100

5**111** Ojos te veanDescripción conceptual, **111** | Descripción tecnológica, **113****6****129** Exhalation-waterResumen, **129** | Concepto, **130** | Descripción tecnológica, **133****7****137** Tell-Tale PianoDescripción, **137** | Desarrollo de la obra, **139****8****149** SwarmsDescripción, **149** | Ensayo de Jill Hardy, **151****9****155** Metáforas para pianos muertosResumen, **155** | Descripción, **156** | Concepto, **157**, Desarrollo y versiones, **158** | Evolución en el desarrollo de la obra, **166** | Relación datos-espectadores, **170****10****185** Objetos resonantesDescripción, **185** | Ecos para locaciones olvidadas, **186** | Primera versión de objetos resonantes, **192** | Segunda versión de Objetos resonantes, **195****11****199** AxialQué es Axial, **199** | Exploración y bocetos, **201** | Propuestas iniciales, **209** | Preocupaciones artísticas en Axial, **212** | Paisaje sonoro submarino, **214** | Preparación de la grabación, **214** | Grabaciones de sonido en el barco Thompson, **222** | Datos en el barco, **224** | Síntesis de audio en la nube, **229** | Formalización de la obra, **230** | Proceso de composición, **230** | Axial estructura y partes, **242** | Montaje, **244****12****259** Helix AspersaResumen, **259** | Concepto, **261**

13

265 Delta 57.00 / -99.00

Resumen, [265](#) | Introducción, [267](#) | Antecedentes, [268](#) | Proceso creativo, [269](#) | Desarrollo de construcción, [272](#) | Desarrollo tecnológico, [276](#) | Composición y sonificación, [279](#) | Proyecto comunitario, [281](#) | Conclusiones y trabajo futuro, [282](#) | Agradecimientos, [284](#)

14

287 GAB

Introducción, [287](#) | 15.2 El sistema gab, [290](#) | El procesador de datos gab 1.0, [292](#) | jmsl y MidiShare, [301](#) | Explicación y transcripción del código fuente, [303](#) | El controlador midi gab, [316](#) | Construcción del controlador midi, [318](#) | Explicación y transcripción del código fuente, [319](#) | Conclusiones, [322](#)

15

325 Interface de música y pintura improvisadora

Motivación, [325](#) | Descripción, [327](#) | Estética del sistema, [331](#) | Tecnología, [332](#) | Hardware, [334](#) | Versiones y proceso de evolución, [336](#) | Idea original, [336](#) | Primera implementación, [337](#) | Segunda implementación, [342](#) | Tercera versión, [357](#) | Conclusión, [367](#)

16

375 Juum

Motivación, [375](#) | Entornos creativos digitales, [377](#) | Descripción, [380](#) | Arquitectura, [381](#) | Código dentro de Juum, [385](#) | Uso de Juum en contextos artísticos, [386](#) | Proyectos realizados con Juum, [388](#) | Conclusiones, [389](#)

17

395 Plurifocal sound controller

Descripción, [395](#) | Tecnología del Plurifocal Events Controller, [396](#) | Fabricación de la electrónica, [399](#) | Fabricación de componentes mecánicos, [405](#) | Aplicaciones del Plurifocal Events Controller, [411](#) | Materiales, [412](#) | Diagramas, [413](#) | Protocolo, [418](#) | Código, [419](#)

431 Conclusión

439 Referencias

*No supe cubrir
mis ojos al realizar
mi viaje hacia al mar
pero, en cambio,
durante mi travesía
pude dejarme caer
ante los brazos
del monstruo sudoroso
y audible y reposar mi oído
en su sonoridad
hasta lograr descubrir
mi propia estatua
y mi asfixia,
estallando
maquinalmente
(siempre en lo audible).*

GABRIELA VILLA *EXPEDICIÓN*

Agradecimientos

Un libro que contiene una revisión de obra de 14 años de trabajo conlleva pensar en todos los colegas, maestros y colaboradores que de manera directa o indirecta han contribuido a la existencia de las obras artísticas aquí descritas. En ese sentido quiero agradecer a todos los profesores que con su ejemplo y paciencia han permitido al autor procesar en sus procesos.

Gracias a Jesús Figueroa de la –en aquel entonces– Escuela Nacional de Música por haber confiado en mí; a la maestra Eva del Carmen por su paciencia en los momentos de iniciación que, ahora sé, son aquellos que necesitan de una guía constante; a Andrés Acosta por la labor y la diligencia de formarme como pianista; a María Teresa Frenk, no sólo por la enseñanza pianística sino por la confianza en permitir la exploración entre el piano y la tecnología; a Mauricio Nader por sus tutorías en ejecución de música contemporánea. En especial agradezco al Dr. Julio Estrada y al Dr. Alejandro Escuer por el inmenso apoyo durante los estudios de licenciatura. Su apoyo dentro y fuera del aula fueron fundamentales para continuar la formación futura. Gracias también a Nick Dikovsky por su inspiración y por sus enseñanzas de programación en la Universidad de Nueva York.

Durante el periodo en el Media Laboratory del MIT, gracias a Tod Machover por abrirme las puertas de su grupo Opera of the Future y gracias a los colegas Tristan Jehan y Mary Farwood por sus enseñanzas y sus asesorías diarias. Gracias también a Joe Paradiso por sus lecciones sobre nuevas interfaces musicales y a John Maeda por sus invaluable clases. Durante el periodo en el Music Technology Group de la Pompeu Fabra gracias a Sergi Jordá por

invitarme al grupo de Interfaces Musicales y a Xavier Serra por permitirme desarrollar mi proyecto dentro de un centro de excelente calidad. Un profundo agradecimiento a Juan Pampin, Richard Karpen, Bruce Hemingway y John Delaney por su guía y enseñanzas durante la etapa en el Center for Digital Arts and Experimental Media, que sin duda fue de gran aprendizaje y un espacio para entender el porqué del arte electrónico.

En estos últimos años, en México, ha sido un placer colaborar con la comunidad de artistas electrónicos. Gracias a Grace Quintanilla por invitarme como director de Tecnología del Centro de Cultura Digital y a Arcángel Constantini por apoyar y fomentar la producción de arte electrónico en México. Gracias por haber confiado en mí. Gracias a Alejandro Ramos por invitarme a formar parte del proyecto de enseñanza en Tecnología Musical del Instituto Tecnológico y de Estudios Superiores de Monterrey. Estoy en profundo agradecimiento con Mónica Benítez y Luz María Sánchez por abrirme las puertas para formar parte del proyecto de construir un espacio para la producción de Arte Electrónico en la Universidad Autónoma Metropolitana.

Sin duda, también estaré agradecido por siempre con mi familia y mis amigos, sin duda son el motor que le da razón al trabajo y fuente constante de inspiración y deseo por realizar proyectos creativos.

Ahora, como profesor de la licenciatura en Arte y Comunicación Digital de la Universidad Autónoma Metropolitana, veo la historia desde el otro lado, como profesor que intenta influir de la manera más positiva en los alumnos. Gracias a los estudiantes de la carrera por su interés y por su deseo de producir propuestas artísticas digitales de calidad. Sin duda este libro es para ustedes.

Hugo Solís García.

Prólogo

En el presente libro Hugo Solís comparte el desarrollo de una gramática propia de la interactividad digital que se expresa en aquella forma del arte contemporáneo que ha sido llamada electrónica, digital o mediática, casi indistintamente. Estas últimas categorías, como nos recuerda el libro, podrían parecer cada vez menos útiles salvo porque reúnen cúmulos de obras que presentan particularidades en su producción, en su circulación y en su análisis e interpretación, particularidades técnicas que resuenan dependiendo de su complejidad en cada momento del proceso del proyecto y en cada dimensión –conceptual, poética, política–. En El proceso en la producción de arte electrónico Hugo Solís nos propone analizar desde la dialéctica que se articula entre el público y la obra las posibles interacciones que emergen de la mediación llevada a cabo a través de sistemas tecnológicos con distintos grados de complejidad; desgrana desde las poéticas presentes en sus propias obras, así como en obras de otros artistas y en resonancia con teorías contemporáneas de los medios, una gramática abierta conformada por los distintos sentidos e intensidades de la interactividad con un propósito claro: organizar y compartir el conocimiento desarrollado a lo largo de más de 20 años de trayectoria.

Los procesos de cada proyecto son analizados desde de sus propuestas iniciales, desarrollo, producción, hasta su interacción con el público, y dichos análisis se convierten en casos que proveen claves importantes tanto para los campos de la producción artística, de la crítica, de la enseñanza, como para el campo museológico. Como artista, investigador académico y docente con una larga trayectoria en cada una de estas prácticas Hugo Solís es

consciente de la gran importancia del desarrollo de un robusto cuerpo de obra, tanto del conocimiento que emerge a partir de dicho desarrollo; de la necesidad de organizarlo, circularlo entre lectores pares, como del placer y la responsabilidad de enseñarlo a las nuevas generaciones, en un campo en donde frecuentemente a falta de documentación y bibliografía se reinventa el hilo negro o se vuelve a estudiar una y otra vez lo que ya otros han aprendido sin dejar rastro. El conocimiento se vuelve tal cuando ha rendido fruto transformando a una persona, cuando en alguien –como diría el neurofisiólogo Francisco Varela– se ha llevado a cabo un cambio en el conocer.

El medio de transmisión del conocimiento en realidad no es muy importante, pero ciertamente el libro ha demostrado una supervivencia en el tiempo que muchos formatos “nuevos” ya no pueden atestiguar. En ese sentido este libro también es singular: sin ser un manual nos provee de manera cercana formas de hacer, posibilidades, aciertos y errores, consejos para la toma de decisiones que se entretajan a lo largo del texto desde una apropiación híbrida como es la de este autor. Ya sea desde el punto de vista de quien recibe una obra en un museo para exponerla, o de alguien que analiza una obra para reflexionar sobre ella, incluso del artista en formación a quien está dirigido este ensayo, el presente trabajo ofrece una enorme cantidad de información técnica, para la mediación con públicos, teórica, sobre derechos de autor, etcétera. En términos pedagógicos esta obra sirve también como referencia para transformar poéticas en didácticas a fin de poder compartirlas en la escuela, espacio que tiene un lugar fundamental en nuestro presente y cuya importancia depende en buena medida de la publicación de documentos híbridos e interdisciplinarios. ■ ■ ■

Tania Aedo

Directora del Laboratorio Arte Alameda.

Introducción

El arte electrónico es, para muchos artistas de las generaciones recientes, una de las ramificaciones expresivas más atractivas y novedosas. Su naturaleza híbrida y su necesidad de diálogo constante con otras disciplinas la convierten en una rama creativa en constante evolución conectada con la sociedad y vigente en sus metodologías y formatos.

Producir arte electrónico requiere de formaciones híbridas y es difícil imaginar a un artista electrónico que no tenga los conocimientos fundamentales en electrónica, fabricación y programación computacional que le permitan ya sea producir íntegramente las obras o, cuando menos, entablar diálogos fructíferos con expertos de cada rama durante los procesos de diseño, conceptualización y desarrollo de la obra artística. Sin embargo, a pesar de los conocimientos variados necesarios para producir obras artísticas electrónicas, muchos de los artistas que se dedican a la producción de arte electrónico se han formado en contextos tradicionales como pueden ser la pintura, la escultura o la composición musical.

Si una basta mayoría de artistas que producen arte electrónico fue formada en contextos tradicionales de creación, ¿cómo es posible que produzca obras con cierta complejidad tecnológica o, en ocasiones, con sofisticados mecanismos tecnológicos? ¿Dónde y cuándo se despertó en cada uno de los artistas el interés por incluir plataformas que estaban fuera de su panorama y espectro tradicional de formación inicial? ¿Cómo se dio el proceso de adquisición del conocimiento necesario para producir obras en plataformas digitales?

¿Cómo fue el proceso creativo en cada una de las obras? ¿Qué hay detrás de cada obra de arte electrónico acabada y mostrada al público como un material concluido?

Si bien es difícil formalizar y presentar un conjunto de reglas o un conjunto de instrucciones que delineen un proceso creativo, existe el consenso de que la composición artística está vinculada a la imaginación y a la libre exploración. La evolución, la transformación y la variación de obras anteriores que la sociedad y los críticos señalan como valiosas obras artísticas pueden ser un buen camino para la creación original. En este sentido, el estudio y el análisis de las obras referenciales es importante y, aun cuando existen excepciones, la mayoría de las obras deriva de trabajos previos.

Mientras que *forma* y *estilo* son conceptos relativamente claros que permanecen estables durante largos periodos de la historia del arte, en el siglo xx tales conceptos se diluyeron por un conjunto de justificaciones complejas sobre la libertad. Los artistas de los años sesenta y setenta rompieron paradigmas y establecieron otras clases de relaciones entre las obras de arte, los espectadores y la tradición. Tal revolución y evolución fue razonable y lógica, pero nosotros en la generación más joven hemos heredado un panorama complejo. Mientras todo parece ser válido en la superficie, en el fondo todavía enfrentamos las inevitables fuerzas de los límites de la percepción, los arquetipos y la memoria social.

Tomará tiempo saber si lo que ahora conocemos como *new media* y como *arte experimental* tendrá en el futuro la validación artística de otras manifestaciones creativas. Mientras tanto, como artistas electrónicos, aprovechamos los tiempos que vivimos y exploramos las posibilidades que ofrece trabajar en múltiples plataformas, con una variedad de tecnologías y creando propuestas artísticas que se encuentran en las intersecciones del arte y lo tecnológico, del arte y la ciencia, del arte y las experiencias sensoriales.

Este libro busca responder las preguntas anteriores desde la perspectiva del creador y con la visión del pedagogo. Por ello, esta inevitablemente se centra en los estudiantes de arte digital y en los investigadores que se ocupan de la arqueología de medios y la creatividad digital. El presente trabajo analiza y explica el proceso de desarrollo en cada una de las obras exponiendo las razones de su creación, el proceso evolutivo de las ideas y las negociaciones entre imaginación e implementación que ocurren durante la producción de una obra artística de corte electrónico.

Aprovechando mi experiencia como autor y productor en el campo de la creación artística electrónica, aquí pretendo exponer las costuras, el andamiaje y el entarimado de cada creación con la intención de mostrar al lector, en especial a los futuros artistas electrónicos, los eventos que hay detrás de las obras: los errores, los bocetos y los intentos fallidos que en muchas ocasiones sus creadores reservan para la intimidad quizá con la buena intención de mantener el aura y la magia que existe al observar sólo la obra terminada.

Cuando iniciamos la formación de artista, ya sea en contextos académicos o de manera libre e intuitiva, el acercamiento que se tiene durante las visitas a museos, galerías y exposiciones en donde se presentan obras terminadas puede ser angustiante y frustrante, en especial cuando aún no se han adquirido los conocimientos y experiencias necesarias para producir una obra. En este contexto el estudiante observa una versión, *la versión* que el artista desea mostrar al público. El artista presenta una obra terminada y es trabajo del estudiante o artista en formación dilucidar el trabajo de producción que hay detrás de la misma.

Ahora bien, es una realidad que la producción de arte electrónico en donde un mismo artista puede explorar formatos y medios variados requiere conocimientos diversos y por tal motivo es normal que el creador delegue parte de la producción a terceros.¹ Una situación posible en el procedimiento de delegar el funcionamiento de una obra a un tercero es que el mismo artista puede llegar a desconocer la mecánica de su mismo trabajo, lo cual complejiza el estudio del proceso creativo.

En este caso particular, todas las obras descritas han sido, además de conceptualizadas y concebidas, diseñadas, fabricadas, ensambladas y programadas por mí, como autor, lo cual permite que exista un conocimiento detallado del funcionamiento de cada obra. Contar con una fuente de primera mano de lo que involucra la creación artística electrónica puede llegar a ser útil para algunos lectores.

Se describe mi trabajo creativo producido desde el año 2001, cuando inicié la producción de obras electrónicas, hasta 2015 cuando culminé una serie de trabajos relacionados con la plurifocalidad.² Se explican las razones de cada pieza y se ofrecen descripciones tecnológicas

¹ Ha sido una agrisadida sorpresa observar que gran cantidad de artistas exitosos en el contexto mexicano delegan una inmensa cantidad del proceso de producción a otras personas.

² Este concepto se presentará varias veces a lo largo del libro ya que ha permeado mucha de la estética de las obras aquí descritas.

y conceptuales de las mismas. El libro integra una colección de obras en diferentes formatos y distintas plataformas entre las que se incluyen el arte cinético, el bioarte, el arte de la tierra y el arte locativo, entre otros.

Es importante mencionar que no pretendemos mostrar las obras desde una perspectiva de valía estética necesariamente –aunque algunas, a mi juicio, tienen fuerza y precisión estética–, por lo contrario, nuestra intención es mostrar de la manera más cruda posible la realidad del proceso creativo, por tanto, es importante hablar de las obras fallidas, de las piezas que no necesariamente lograron el impacto esperado o de aquellas cuya materialización tecnológica quedó distante de la especulación artística. Un objetivo implícito es mostrarles a los estudiantes de arte electrónico que no todo dentro de los procesos creativos es éxito y que es importante saber convivir con una realidad en donde algunas obras llegan más lejos que otras.

Tener una *historia no editada* de un recorrido artístico puede ser también interesante para algunos lectores ya que les da la oportunidad de revisar pensamientos recurrentes entre todas las obras, o por el contrario, ideas que no continuaron desarrollándose. También, y como es lógico, se podrá constatar cómo algunas ideas evolucionan entre una obra y la siguiente. Se pueden observar *giros de tuerca* o incluso estéticas que fueron probadas, pero no continuaron su proceso de desarrollo. Hablar de dichos procesos no es necesariamente fácil, pues una revisión del trabajo conlleva a revisar una historia no editable del pasado. Visto a la distancia algunas decisiones pudieron ser diferentes y otras, por el contrario, fueron selecciones pertinentes. Lo inevitable es que las obras aquí descritas son la historia *real* del trabajo que el creador ha desarrollado en 14 años y dicho recorrido, como ya hemos mencionado, puede ser útil para la comunidad de artistas electrónicos, para los estudiantes y para los revisores de la estética del arte electrónico.

Lo único que en este momento y después de revisar los materiales aquí seleccionados podemos observar es que el sonido es una constante y que con dicho medio se busca expresar ideas de diversa índole encapsuladas en lógicas variopintas. A mi juicio, el sonido y la programación computacional son una constante en el trabajo presentado. Con estos dos elementos se busca que, al contemplar una obra, el público espectador reflexione y se formule cuestionamientos. El uso del sonido en las obras que aquí presentamos se desarrolla

como una evolución natural e inevitable de una curiosidad primaria por la música y por el propio sonido.³

Espero, de la manera más honesta posible, que mostrar los aciertos y los errores a lo largo de estos años como creador de experiencias electrónicas sensibles sea útil para los lectores interesados en el campo de la producción de arte electrónico. Espero que este documento sirva a mis alumnos de la licenciatura en Arte y Comunicación Digitales de la Universidad Autónoma Metropolitana como un ejemplo, pero no de lo que se debe hacer, sino como un escaparate de ejemplos que les permita rastrear un proceso de producción con aciertos y errores. Sobre todo, que les permita desmitificar lo antes posible la imagen del creador/mago y que se inicien en la búsqueda de sus propias preguntas y soluciones sin temor, y con la certeza de que con sensibilidad y trabajo constante podrán producir sus obras y contribuir a la paleta expresiva dentro del campo del arte electrónico.

EL LIBRO

Este texto está dividido en cuatro partes, la primera presenta el trabajo realizado de 2001 a 2010 e incluye trabajos producidos durante el periodo de formación durante la licenciatura en Piano en la ahora Facultad de Música de la Universidad Nacional Autónoma de México, durante la maestría en Arte y Ciencia en el Media Laboratory del Massachusetts Institute of Technology; durante la maestría en Informática y Comunicación Digital del Music Technology Group de la Universidad Pompeu Fabra y durante el doctorado en Arte Digital y Medios Experimentales de la Universidad de Washington. En esta primera parte se incluye una sección con los trabajos de música y escena que en un principio fueron actividades constantes en mi práctica artística y que sin duda impactan la producción de obras posteriores.

³ Mi formación inicial fue como ejecutante de piano. Es muy probable que de haber existido una carrera de arte electrónico en la época de mi formación de licenciatura la habría estudiado. En ese sentido debo agradecer enormemente a los profesores que tuve en la universidad, que detectaron mi inclinación por las manifestaciones electrónicas y me dieron todo el apoyo y libertad posibles para iniciar proyectos de desarrollo tecnológico muy al principio de los estudios de piano.

La segunda parte se conforma del trabajo entre 2011 y 2015. En esta sección se presentan tres proyectos de gran envergadura. Si los trabajos iniciales eran pequeños, ahora cada proyecto requiere investigaciones profundas y desarrollos extensos.

La tercera parte muestra varios proyectos tecnológicos desarrollados en diferentes momentos. La intención no es mostrar las aportaciones tecnológicas únicamente, sino dejar una evidencia de lo que implica el desarrollo tecnológico aplicado al arte. Algunos de estos desarrollos siguen vigentes y otros han quedado obsoletos. Es interesante revisar trabajos los cuales en algún momento requirieron cientos de horas de desarrollo y que ahora son sobrepasados por proyectos comunitarios de código abierto.

Muchos de los trabajos que presentamos son previos a la era de Arduino y de Processing, por citar algunas de las herramientas que han sido adoptadas y asimiladas por la comunidad de arte electrónico. Es un gusto saber que la complejidad de muchas herramientas se ha vuelto de uso cotidiano; sin embargo, hay que aclarar que su mera existencia no es garantía de una apropiación adecuada por parte de, por ejemplo, los estudiantes. Las bases de programación, de pensamiento lógico, de electrónica y mecánica básica siguen siendo un requisito indispensable para poder utilizar dichas herramientas.

Es importante mencionar que partes de este libro se han obtenido de textos escritos, nunca publicados, en diversos periodos de tiempo. En ocasiones, al terminar una obra y hacerle difusión se escribían materiales que se publicaban en páginas web o que se anexaban como descripciones de proyectos al aplicar a becas o apoyos de producción. Algunos de estos materiales han sido dejados intencionalmente en su forma original –haciendo las notas y aclaraciones pertinentes– independientemente de su vigencia, ya sea como planteamiento estético o de su caducidad tecnológica. Considero importante que dichos materiales aparezcan de esta manera para poder apreciar la evolución de los planteamientos artísticos y la evolución en el desarrollo tecnológico.

Por último me gustaría decir que pertencí a la última generación de personas que creció sin internet. Hay cierto dejo de nostalgia en la manera en que fuimos formados y en la que adquirimos la información que necesitábamos para producir nuestro trabajo. Hoy en día el acceso a la información no es un problema y sin duda este paradigma ha cambiado la forma de producir materiales y obras artísticas digitales. Espero que este repertorio de experiencias,

como expliqué anteriormente, le sirva a los lectores –en particular a los estudiantes de arte electrónico– para revisar un procedimiento creativo y que puedan tomar lo que sirva de él y no repetir los errores que sin duda se han presentado en el proceso.

Notas en relación al código fuente

Como se podrá ver, este libro incluye una gran cantidad de código fuente. En los años ochenta, antes de internet y cuando la producción de CD seguía siendo costosa, era común encontrar libros que contenían este tipo de información. En nuestros días no es indispensable y en muchas ocasiones no es deseable, ya que tenemos acceso a este material por otros medios. Me doy la licencia de incluirlos, así como incluyo una gran cantidad de gráficos, como una mecánica para hacer un claro manifiesto de que el código es pensamiento y tiene un reflejo visual y textual. Tener un impreso en donde conviven código, texto e imagen me resulta importante para explicarles sobre todo a los estudiantes que en la producción artística electrónica las fronteras de todos los medios está desdibujada. Aún así es importante mencionar que todo el código está montado en un repositorio abierto en GitHub al cual se puede acceder desde mi página web en <http://hugosolis.net>. Si la intención del lector es manipular, extender o mejorar los códigos tiene la posibilidad de hacerlo directamente desde el repositorio digital.⁴

Comentarios en relación a las notas al margen

A lo largo del documento se hace un uso frecuente de notas al margen. Es importante mencionar que aquellas que están numeradas son citas y notas a pie de página, mientras que las notas que no se encuentran numeradas son apostillas relacionadas con ideas y pensamientos secundarios al texto principal. ■ ■ ■

⁴ En 1987, cuando tenía 11 años, compré en el puesto de periódicos una revista que contenía el código fuente en el lenguaje Basic para hacer el juego de Gato. Tenía aproximadamente 1 200 líneas de código que copié manualmente a una computadora Apple IIc sin saber programar. Nunca funcionó dados los errores que contenía el texto y los errores de dedo. Moraleja, si el lector va a utilizar el código de este libro, que lo descargue del repositorio.

Arte interactivo y participación del público como agente colaborador

INTRODUCCIÓN

Una parte importante de mi trabajo contiene mecánicas interactivas en las que el público funciona como un agente que *activa* la obra en muy diversas capacidades, niveles y relaciones. El tipo de relación entre obra y público varía en cada caso, pero será valioso establecer un marco de pensamiento en relación a este tipo de manifestaciones. En este capítulo se presentan ideas relacionadas con el arte interactivo y la participación del público dentro del arte electrónico.¹

El arte interactivo, sus propiedades y sus definiciones son nuevos en relación a la historia del arte en su conjunto. La participación del público como agente colaborador obedece a factores entre los que se incluyen la obra abierta, las teorías de la información y, sobre todo, la era digital. Más que un tipo de arte es una propiedad posible dentro de la expresión artística muy visible dentro del campo del arte electrónico y de los *nuevos medios*.

La interactividad en el arte puede ser estudiada a través de los niveles de relación e impacto de las acciones de los espectadores dentro de la obra, por las relaciones que se generan entre ésta y los participantes y, paralelamente, por las relaciones que se generan entre los espectadores. Conceptos como tiempo, privacidad, autoría, entorno y espacio público juegan

¹ Este texto fue escrito como un ensayo independiente para el examen de oposición en la Universidad Autónoma Metropolitana.

papeles importantes dentro del arte interactivo. Éste, en el contexto del arte electrónico requiere plataformas y herramientas para su producción. En la actualidad se utilizan diferentes tipos de sensores para obtener información de los espectadores. Existen además herramientas, software y tecnologías que han sido adoptadas por la comunidad del arte electrónico, que permiten su producción, distribución y enseñanza.

Dado que la emulación, el modelado y el análisis de datos pueden ser mecanismos que expanden la expresividad de las obras interactivas modernas, el *mapeo* de datos y la adopción de técnicas de inteligencia artificial y aprendizaje de máquina son herramientas técnicas que pueden ser aprovechadas en su producción.

Este capítulo aborda el campo del arte interactivo y la participación del público como agente colaborador dentro del contexto del arte que se produce actualmente. Se centra, particularmente, en el campo del arte electrónico en donde el uso de herramientas digitales y tecnológicas potencia dichas interacciones y en donde la acción de los espectadores es evidente y directa. Se centra en dicho campo dada su importancia dentro de la licenciatura en Arte y Comunicación Digitales que se imparte en la Universidad Autónoma Metropolitana, unidad Lerma, y dada mi experiencia como creador y pedagogo en dicho campo.

Este capítulo está dividido en siete secciones, cada una de las cuales aborda temas específicos que en su conjunto ofrecen una descripción del estado del arte dentro de dicha temática. No pretendo abordar todas las aristas y perspectivas existentes, pero sí ofrecer un panorama general del arte interactivo y de la participación del público como agente colaborador, ya sea como detonador, copartícipe o cocreador. Cuando lo considero pertinente hago referencia a trabajos icónicos, menciono obras realizadas en México y, en ocasiones, presento trabajo de mi autoría.

CONTEXTO HISTÓRICO Y ARTÍSTICO

El uso de recursos alternos para la producción de arte no es nuevo. En 1834 William George inventó el Zoótopo y con ello estableció una manera de generar imágenes dinámicas de forma controlada. En 1958 Wolf Vostell incorporó televisiones en su trabajo artístico *The Black Room Cycle*. De la misma manera, con la invención de los sistemas digitales surgieron

también propuestas que adoptaban dicha novedad para generar y materializar obras creativas que, difícilmente, podrían ser creadas sin dichos recursos en aquel entonces novedosos. En 1956 se generó el primer dibujo mediante una computadora militar y, en 1960, Desmond Paul Henry inventó The Henry Drawing Machine. En 1962, Michael Noll programó una computadora digital para generar patrones visuales con fines artísticos (Tribe y Jana, 2006).

A lo largo de la segunda mitad del siglo xx y de manera gradual, la adopción de nuevos recursos y nuevas estrategias creativas y tecnológicas dio pie a manifestaciones expresivas que adoptaban herramientas y recursos de reciente creación. Estas adopciones se dieron principalmente dentro del marco de las propuestas experimentales y exploratorias. Ejemplos notables son los trabajos de Nam June Paik, Frieder Nake, Leslie Mezei, Georg Nees, Michael Noll, John Whitney y Charles Csuri quienes son considerados los primeros artistas digitales (Paul, 2008).

Al mismo tiempo que se daban las propuestas artísticas personales podemos ver que a partir de los años setenta diversos centros e iniciativas se fueron creando gradualmente para generar diálogos entre el arte, la ciencia y la tecnología. Así pues no es de extrañar, por citar un ejemplo, que en octubre de 1966 se haya llevado a cabo una extensa colaboración entre ingenieros de Bell Laboratories y un grupo de artistas el cual culminó con una serie de eventos llamados *9 Evenings: Theatre and Engineering* (Hultén y Konigsberg, 1966) en donde participaron artistas como John Cage y David Tudor, e ingenieros como Max Mathews.

Las iniciativas creativas de aquellos tiempos se incorporaron gradualmente de manera formal a las instituciones académicas y podemos ver, por ejemplo, que el Massachusetts Institute of Technology creó, en 1967, el Center for Advanced Visual Studies, orientado a buscar innovación en el campo de las artes visuales, del entorno, la danza y los nuevos medios.² Este pensamiento proliferó y en la actualidad nos encontramos con planes y programas de estudio en diferentes partes del mundo en donde la creatividad digital desempeña un papel prioritario.³

² Es interesante observar cómo el CAVS y el programa de Visual Arts (VAP), creado en 1989, se unificaron en 2009 para generar el programa de Arte, Cultura y Tecnología (ACT). "History of act", MIT.

³ El programa de DXARTS del Center for Digital Arts and Experimental Media de la Universidad de Washington, en donde obtuve mi doctorado, fue de los primeros programas de arte electrónico a nivel licenciatura y desde 2004 ofrece un doctorado de orientación práctica en arte digital y medios experimentales.

Contexto mexicano

Dentro del contexto nacional mexicano, en 1994 se creó el Centro Multimedia dentro del Centro Nacional de las Artes y desde entonces ha sido un referente y un lugar de investigación del arte electrónico. En 2000 se creó el Laboratorio Arte Alameda que aloja exposiciones de corte tecnológico y electrónico. Más recientemente, en 2012, se creó el Centro de Cultura Digital que, si bien no es un lugar de corte artístico, es un espacio para la creación, la enseñanza y la apropiación social de nuevas tecnologías.

En el entorno académico, la Universidad Autónoma Metropolitana cuenta con un grupo de investigadores que dentro de la división de Ciencias y Artes para el diseño han generado iniciativas interesantes como es el caso del Jardín Sonoro de la UAM Xochimilco. La Universidad Nacional Autónoma de México y el Politécnico Nacional han tenido iniciativas interesantes mediante diplomados, congresos, seminarios y grupos diversos. En junio de 2018 se inauguró la carrera de Tecnología Musical en el campus de Morelia de la UNAM que, por su naturaleza, tendrá un área de desarrollo tecnológico en el área del arte sonoro digital, que evidentemente tiene una conexión directa con el arte electrónico e interactivo.⁴ Algunas instituciones particulares como el Tecnológico de Monterrey, SAE y CENTRO cuentan con carreras que ofrecen cursos aislados de interactividad en el arte.

En el sector de iniciativas sociales no cabe duda de que la creación de recursos de acceso libre como Arduino, y los costos cada vez menores de herramientas tales como las impresoras 3D y las cortadoras láser han ayudado a cimentar una cultura de *makers* y *hacedores*. El Maker Lab de la Ciudad de México, el Laboratorio de la Ciudad de México, la Fábrica Digital Rule, entre muchos otros, son espacios de creatividad tecnológica que si bien no están orientados al arte pueden ser semilleros para la creación de grupos interdisciplinarios de corte creativo que, detonarán, seguramente, propuestas artísticas de valor e importancia.

Los new media como punto de partida

Con el paso del tiempo, el compendio de esfuerzos y propuestas personales de artistas y creadores, así como la tendencia hacia una cultura digital, se consolidaron lo suficiente para

⁴ “Carrera, música y tecnología”, *La Jornada*.

que emergiera el término de *new media* o nuevos medios para referirse, en su conjunto, a toda la gama de expresiones artísticas que se encontraban fuera del marco tradicional del arte y que tenían como elemento unificador la búsqueda de paradigmas expresivos, utilizando los recursos tecnológicos claramente asociados con el mundo electrónico y digital (Kwastek, 2013).

En este sentido, Lev Manovich plantea que, independientemente de lo heterogéneo del medio, las propiedades de esta expresión radican en su representación numérica, su modularidad, su automatismo, su posibilidad de variabilidad y finalmente en su transcodificación entendiendo esta última como la capacidad de las obras de actuar como traductores bidireccionales entre la realidad y los datos (Lev, 2001). Manovich también nos habla de interactividad, pero la mantiene separada por motivos que se explican más adelante.

Dentro de los nuevos medios nos encontramos con temáticas tales como el bioarte, el arte telemático, el locativo, el arte en videojuegos, el generativo, el de la información, el *net art*, el arte kinético, el *software art*, el arte virtual y el arte algorítmico, entre otros (Wilson, 2002). Evidentemente, estas categorías son meramente clasificaciones abstractas que intentan organizar una gama de expresiones en un espectro continuo de manifestaciones y recursos diversos. Una misma obra puede pertenecer a varias áreas o simplemente no estar claramente definida. Al mismo tiempo, términos como nuevos medios, arte electrónico y arte digital son conceptos difusos en donde se traslapan características y terminologías. Como nos dice Kwastek (2013):

Las designaciones en el subtítulo de arriba⁵ no se originaron como auto-definiciones provocativas hechas por grupos de artistas (como en el caso del “futurismo” y el “Nouveau Réalisme”), o como etiquetas inadecuadas (como en “gótico” o “impresionismo”). Todos los términos fueron introducidos como clasificaciones ostensiblemente objetivas basadas en características técnicas. Continuamente, sin embargo, estos términos se han vuelto equivalentes dentro de los intensos debates de las nuevas formas de arte que ellos designan y son, consecuentemente, criticados en su lugar.⁶

⁵ Se refiere al nombre del capítulo: “New Media Art, Computer Art, Digital Art and Interactive Art”.

⁶ Traducción del autor.

El problema taxonómico de las categorías y clasificaciones se complejiza aún más si consideramos que el uso de las herramientas electrónicas y digitales que ayudaron a gestar un pensamiento artístico propio es, en nuestros días, una constante de uso cotidiano independientemente de si el formato es tradicional o no. Fotógrafos, cineastas, músicos, escultores, entre un sin fin de creadores, contemplan recursos digitales en la conceptualización, creación, producción o difusión de las obras. Es difícil pensar en un arte puramente analógico sin remitirnos a una propuesta que intencionalmente quiera hacer una reflexión de la evolución tecnológica. Evitar el uso de herramientas digitales es, consciente o inconscientemente, un manifiesto por parte del creador quien no puede estar libre del contexto en el que vive y produce su trabajo.

Por tanto, y cayendo un poco en la obviedad, es evidente que no por el simple hecho de utilizar recursos electrónicos una propuesta deba considerarse arte electrónico, más aun si incluimos al arte digital dentro del arte electrónico. El arte electrónico ha generado una dialéctica propia, contiene lógicas particulares y da énfasis a ciertas relaciones entre la dinámica de observación, la forma y el uso de los recursos tecnológicos de las cuales hablaremos continuamente a lo largo del texto.

ARTE INTERACTIVO, UNA CLASIFICACIÓN IMPRECISA

Considerando lo anterior, ¿por qué hablar de un *arte interactivo* si el término es poco preciso y difuso? Lo más acertado sería postular que la interactividad es una propiedad posible de los nuevos medios y que una gama de expresiones de dicho campo se apropia de ella de manera directa. En otras palabras, la interactividad no declara un tipo particular de arte; más bien, la interactividad es una propiedad del arte exponente de manera más directa y más evidente con la incorporación de las tecnologías de la información y comunicación digitales.

Manovich nos habla del *mito de la interactividad* para establecer la ambigüedad del término y nos dice: “Así como con *digital*, prefiero evitar la palabra interacción en este libro sin calificarla, por la misma razón; encuentro el concepto demasiado amplio para ser verdaderamente útil.

Sería importante manifestar que, evidentemente, desde una perspectiva más genérica y filosófica todo el arte presenta un cierto grado de *interactividad* con el espectador y con ello, probablemente, todos los creadores aspiran a una participación *activa y creativa* por parte del observador, ya sea imaginativa, conceptual o tangible. También será importante mencionar, como explica Manovich Lev (2001), que: “En los años sesenta, continuando donde el Futurismo y el Dadaísmo se quedaron, nuevas formas como los *happenings*, el *performance* y la instalación convirtieron el arte explícitamente participativo; una transformación que, de acuerdo a algunos teóricos de los nuevos medios, preparó el terreno para las instalaciones interactivas computacionales que aparecieron en los ochentas”⁷

Será importante explicar que para nuestros fines hablaremos de interactividad en el sentido adoptado dentro del discurso del arte contemporáneo. Es decir, arte que requiere de un *input* directo por parte de los espectadores para poder considerarse una obra terminada y funcional. Hablamos de obras no lineales en donde al menos uno de los elementos resultantes, comúnmente observable, dependerá de la información que proviene del espectador, directa o indirectamente, ya sea presencial, o remotamente en el momento de la observación, o anticipadamente. Dicha generación de materiales por parte del público participante es de una muy variada gama de posibilidades como explicaremos a continuación.

NIVELES DE COLABORACIÓN E INTERACTIVIDAD

¿Cómo repercuten las acciones de los espectadores en la obra artística? ¿Qué papel juegan sus decisiones dentro de la obra y cuál es su rango de acción? Eso dependerá del marco previo establecido por el creador pudiendo ser éste muy restringido y limitado o, por el contrario, extendido para repercutir de forma significativa en la obra. La clasificación es únicamente un marco simplificado con fines de organización.

El espectador como detonante

En este caso, la forma y la estructura de la obra artística están establecidas de antemano por el artista. Los materiales están preestablecidos y el espectador actúa únicamente como

⁷ Traducción del autor.

detonante de bloques estructurados con anterioridad. Es, por tanto, la forma más simple de interacción que, si bien funciona en algunos contextos, es también el problema en muchas de las obras electrónicas debido a que la relación con el público tiende a ser llana y superficial. Sin embargo, suelen ser también obras de fácil lectura y su predictibilidad gusta a los grandes públicos. En la pieza de 2008, *When laughter trips at the threshold of the divine* de Kim Beck y Osman Khan (Khan, 2017) la presencia de los paseantes se utiliza para activar una puerta eléctrica en un parque. Aunque hay sutilezas conceptuales y referencias históricas interesantes, la interactividad es muy primitiva y los recursos tecnológicos en su dialéctica son precarios.

El espectador como cocreador

En un segundo nivel de interactividad nos encontramos con obras artísticas en donde el público funciona como cocreador de las mismas. Más arriesgadas son las obras en las que se espera un compromiso y un nivel de involucramiento más profundo con ellas. El espectador debe suministrar materiales que la obra transforma, mezcla, altera y manipula según la lógica del trabajo. Por tanto, el nivel de variación de la obra es mayor y dependerá de los límites preestablecidos. Técnicamente hablando, el nivel de ruido, entendido éste como un material que se encuentra fuera de la lógica de la obra, puede significar un problema y, por tanto, el nivel de análisis de los materiales de entrada debe realizarse en un nivel al menos primario. En la pieza *vtol: oil* de Dmitry Morozov (2015) el artista les pide a los participantes colocar un dispositivo electrónico dentro de unas prensas hidráulicas que destruyen el aparato al tiempo que registran los sonidos de la destrucción.⁸ La selección del aparato y la posición en que el participante lo coloca en la prensa dará resultados sonoros diferentes y es hasta ese punto adonde llega el rango de acción del participante. En estas propuestas de interacción el participante es guiado y constantemente dirigido durante la navegación y el recorrido de la experiencia.

⁸ Entre otras cosas, el proyecto lleva a un territorio físico los cada día más radicales conceptos del sonido craquelado (Kelly, 2009).

El espectador como creador

En otro nivel de interactividad nos encontramos con obras en donde el marco establecido por el creador es muy amplio y se espera que la obra sea una agregación controlada de materiales creados en su totalidad por los espectadores. Son propuestas complejas con altos niveles de ruido y situaciones no controladas donde existe poco arbitraje por parte del creador principal. Por lo anterior, son las propuestas en las cuales el análisis de los materiales y su clasificación, estudio e interpretación suelen ser importantes. La pieza *Reactable* (Jorda, 2005) es una obra (instrumento musical) audiovisual donde el espectador participa como creador y ejecutante último de la misma.⁹

INTERACTIVIDAD EN CONTEXTO

Hasta este punto hemos abordado la interactividad como una propiedad posible en el arte electrónico. Sin embargo, se ha tocado el tema desde una perspectiva un tanto teórica y abstracta. La realidad es que los conceptos, fundamentos teóricos y marcos conceptuales se decantan en obras artísticas materiales, palpables y apreciables en el mundo de los sentidos. Las obras terminan teniendo cuerpo y forma y los creadores han de mediar constantemente entre sus ideas e imaginarios y la materialización de los contenidos que, en la esfera electrónica, suelen permear incluso las propuestas estéticas.¹⁰

Como vimos con anterioridad, la interactividad es sobre todo una propiedad del arte adoptada fuertemente en el arte electrónico, que permite diálogos directos con el público y los espectadores. Como tal, tiene manifestaciones diversas en función de las características de la propuesta artística determinada. A continuación, presentamos relaciones con-

⁹ Como miembro del Grupo de Sistemas Interactivos del Music Technology Group tuve la oportunidad de vivir el proceso de creación del *Reactable* desde sus inicios y ver la evolución de este trabajo. Fue una experiencia altamente pedagógica observar la maduración de una propuesta desde una idea de Sergi hasta un producto que, sin duda, marcó un hito en la historia de la creación de instrumentos musicales.

¹⁰ El medio es tan importante en el arte electrónico que existen constantes debates sobre su caducidad, su preservación y las dificultades a las que se enfrentan museos, galerías y coleccionistas para mantener obras electrónicas en sus colecciones.

textuales bajo una lógica de *usabilidad*, relacionando la interactividad con otras propiedades de las obras electrónicas.

Interactividad individual

Algunas manifestaciones artísticas han de ser experimentadas en el contexto de lo individual. En este sentido las obras de *netart* que tienen salida en un navegador de internet y con las cuales se interactúa desde la interface de la computadora tradicional son las más comunes. Un ejemplo es la obra artística que *existe* en la web, con la cual el espectador juega, experimenta y descubre. La obra es una encarnación única cuya existencia dura mientras el espectador la tenga en su pantalla. En estas propuestas cada obra es independiente del resto de las encarnaciones y sólo hay conectividad entre un espectador y la obra interactiva. Ejemplos diversos se pueden encontrar sobre todo en el *game art* y en las primeras manifestaciones del *netart*. En el juego *La intrusa* la artista Natalie Boochin usa el videojuego como una reflexión sobre una metáfora de la narración misma basada en un cuento de Borges del mismo nombre (Paul, 2008)

La colectividad creadora

Existen, por lo contrario, obras que funcionan bajo la lógica colectiva, obras que están pensadas para completarse en observaciones y aportaciones gregarias. Independientemente de si los espectadores están en el mismo punto físico o de si están distribuidos en espacios reales o virtuales diversos, en estos casos la obra requiere más de un espectador, pues las lógicas de interacción son colaborativas.¹¹

Habría que resaltar que la información de los medios es, en cierto sentido, una colectividad mediada por un canal de información. En estos casos, la colectividad es agregada y filtrada previamente, pero el observador está ante un conjunto de materiales producidos por una colectividad global que no necesariamente reconoce que sus acciones son parte de la obra. *Listening Post* (Colson, 2007) es una maravillosa obra de 1999 donde diversas fuentes

¹¹ De reciente impresión, el libro de Mónica Benitez et. al. es un valioso trabajo que gira en torno a la “reflexión alrededor de manifestaciones artísticas recientes que utilizan distintos recursos tecnológicos para incluir a un público participante y activo como uno de los ejes rectores en el desarrollo de las mismas”.

de noticias y chats se integran en un visualizador audiovisual compuesto por una matriz de pantallas de texto. La colectividad está en la fuente generadora de materiales. No se espera ni analiza nada del espectador directamente, pero siendo el espectador parte de las noticias globales, es parte de los datos que, tentativamente, pueden existir en las fuentes de información que se utilizan.

Interactividad en el espacio público

Una de las manifestaciones más atractivas para el público general es la obra interactiva en espacio público. Pudiendo ser o no obras *site specific*, la interactividad en el espacio público es un tema fascinante, ya que, de cierta manera, la obra se convierte en un mecanismo de apropiación y cohesión social. De forma intencional o no, la obra obliga a los espectadores a reconocer al otro y a navegar espacios físicos particulares. Las acciones del público en relación a su impacto en la obra pueden o no ser en tiempo real y en ocasiones la masificación impide analizar la contribución individual, sin embargo, pareciera ser que esto no es problema ya que las obras se convierten en organismos colectivos de autoría indeterminada. Un ejemplo asombroso en el contexto mexicano lo encontramos en la obra *Vector Elevation*[¹] de Rafael Lozano-Hemmer. En esta obra, una configuración de luces aéreas generaba patrones cambiantes sobre el Zócalo de la Ciudad de México a partir de los modelos que el público realizaba en un portal de internet.

Interactividad, privacidad y anonimato

El hecho de obtener algún tipo de información del espectador implica que la obra interactiva se *involucre* con el participante por lo menos en un nivel primitivo y, por tanto, se da una negociación de privacidades. La obra artística se deja apreciar bajo la condición tácita o implícita de que el espectador haga público algún elemento de su ser. “Te dejo que me goces, pero antes dime algo de ti” pareciera plantear en cierto sentido la obra interactiva.

Es sabido que cientos de portales de internet analizan la posición del ratón del visitante y con ello infieren sus intereses y atención (Baker, 2009). Con esto se pretende explicar que incluso la ingenua obra artística que demanda sólo la posición x y y de un cursor establece una

relación compleja de privacidad. Imaginemos entonces la diplomacia y la política de relaciones que se dan en obras complejas como la pieza *Sanctum* de Juan Pampin y James Coupe en donde los rostros de los participantes son clasificados en edad, raza y género para asociarlos con visitantes similares tanto reales como equivalentes demográficos existentes en Facebook y otras redes sociales digitales (Pampin, 2016).

Hay obras artísticas en donde la interacción no necesita de un participante específico. Se requiere su información y sus datos, pero no su identidad. Ante la problemática de la privacidad surge evidentemente el anonimato. En este caso, la interacción es anónima y la obra funciona en su totalidad. En otros casos se busca la identidad. La obra requiere el *rostro* que acompaña a las acciones; en otros casos la autoría del dato no se puede ocultar pues, por ejemplo, se ve a la persona que acerca su cuerpo al sensor para detonar un evento particular. En la pieza *Pockets Full of Memories* (Wands, 2006) el artista George Legrady requiere que cada visitante del público contribuya con la imagen de algún objeto que lleve en sus bolsillos al momento de entrar a la instalación. El archivo crece a lo largo de la exhibición y los visitantes van dejando rastros de su identidad sin problemas de privacidad porque tácitamente se sabe que dicha información es parte central del discurso narrativo.

Interactividad y tiempo

La interactividad puede darse, hablando del tiempo, de diversas formas y con gamas variadas. Puede ser en tiempo real, en donde la alimentación de uno o varios espectadores se observa de manera inmediata, como en el caso de varias de las obras de Golan Levin quien en *Dialtones: A telesymphony* (2010), por ejemplo, accede a los teléfonos celulares de una audiencia de 200 personas y a distancia controla los sonidos al momento de la ejecución.

Por el contrario, la interactividad puede estar diferida en el tiempo. Las acciones de los espectadores no repercuten de forma instantánea en la obra, pero sus acciones y contribuciones son registradas para una agregación que permite a los nuevos observadores tener conciencia de las acciones anteriores. Observemos, por ejemplo, el proyecto *canal*ACCESIBLE* de Antoni Abad (2006) en donde la agregación de materiales diferidos a lo largo del tiempo es la que va produciendo el mapa virtual de todos los lugares inaccesibles para personas con discapacidad motriz en una ciudad.

Interactividad y distancia física

Evidentemente, uno de los impactos importantes en el uso de redes fue la posibilidad de generar obras cuya interactividad y participación estuviera repartida geográficamente. Conocidas en su conjunto como *telemáticas* incorporan desde su esencia el flujo de datos. Se han logrado obras de corte global tanto en el ciberespacio como en la realidad física o en mezclas de ambos territorios. La pieza *Telematic Dream* (Shanken, 2009), de 1992, es un ejemplo icónico en donde se aprovechan las telecomunicaciones para poner de manifiesto el concepto mismo de locatividad y mediación geográfica. En esta obra se logra y se juega con la intimidad de dos personas en espacios geográficos diferentes, pero que comparten la misma cama al menos de manera virtual. En cada punto se proyecta el video en tiempo real del otro individuo.

Interactividad y su relación con los medios tradicionales

Como hemos visto, la interactividad es una propiedad de determinados tipos de arte y se manifiesta de manera evidente en el arte electrónico, digital y experimental. Dentro del arte electrónico, la interactividad juega un papel preponderante donde las expresiones artísticas son y existen gracias a las nuevas tecnologías. El arte telemático y el *netart* son expresiones nacidas en el seno de la cultura digital, pero ¿qué rol juega la interactividad dentro de las artes tradicionales y cómo ha influido la interactividad digital en la evolución de las artes tradicionales? El concepto de obras artísticas que cambian y se manipulan de acuerdo con situaciones ajenas a la obra ha sido tratado a profundidad por teóricos, semióticos y filósofos. En la música, por ejemplo, se ha dado paso a la música algorítmica y a la construcción de arte sonoro interactivo cuyos materiales, independientemente de ser ejecutados por medios digitales o con instrumentos acústicos, son generados en tiempo real y en función de valores externos y cambiantes.

TÉCNICAS Y RECURSOS PARA LA PRODUCCIÓN DE ARTE INTERACTIVO

Propuestas estéticas que están ligadas al desarrollo tecnológico, electrónico y digital están fuertemente vinculadas a las herramientas que se utilizan para su creación. En algunas ocasiones el diálogo entre la estética y la técnica es parte del proceso creativo y en ocasiones los compromisos y negociaciones para resolver las obras artísticas permean los procesos creativos en sí mismos. En este sentido, es importante hacer una revisión de las tecnologías y herramientas que actualmente se utilizan en la producción de propuestas interactivas que permiten incorporar la actividad y la participación del público.

Plataformas analógicas

Antes de enfocarnos en el tema central de las artes electrónicas sería pertinente hacer referencia a un subgrupo dentro del arte interactivo, que no se encuentra vinculado al mundo digital. Se trata de obras cuyos resultados están influenciados por las acciones del público sin necesidad de un traductor digital ni de análisis numérico. Su estudio profundo está fuera del alcance de este libro, pero es importante mencionar la importancia de los trabajos colaborativos, participativos y comunitarios en los que los participantes realizan actividades que, en su conjunto, definen la propuesta artística. Dichos trabajos están enmarcados dentro de las artes alternativas y los mantendremos al margen de este capítulo con la finalidad de centrarnos en nuestro tema de estudio, que es el uso de estructuras electrónicas y procesos digitales.

Plataformas electrónicas

En las piezas de arte electrónico es cotidiano encontrarse con mecanismos que permiten la manipulación de datos codificados digitalmente. Evidentemente, el uso de computadoras tradicionales es lo ordinario. Sin embargo, será importante detallar el entorno, especialmente por el momento tecnológico en el que nos encontramos en donde el cómputo físico, internet y los sistemas embebidos están diluyendo el significado tradicional de la computadora.

Las computadoras de escritorio y las computadoras portátiles han sido el sistema más común de trabajo. El sistema operativo y el software utilizado son, por tanto, parte de la

plataforma de la obra. Los dispositivos tradicionales de entrada, tales como el teclado, el ratón, el micrófono, la cámara y equipos extendidos como controladores de videojuego se convierten fácilmente en mecanismos para obtener la información del espectador o del participante de forma convencional. La señal de la tarjeta de video en su salida a pantalla o proyector, y la de audio en su salida a bocinas, equipos de audio o audífonos se convierten en los mecanismos de salida.

El deseo de incorporar una *fisicalidad* palpable en el arte electrónico generó la adopción de microcontroladores como puentes entre el mundo digital y el entorno real. Los microcontroladores son circuitos integrados que permiten definir la rutina de acciones a ejecutar y cuentan con registros digitales interconectados a *pines* físicos, lo que permite generar o cuantificar variaciones de voltaje que pueden ser aprovechados en sistemas electrónicos externos. Los microcontroladores pueden funcionar de manera autónoma, o conectados a la computadora y, en ambos casos, son programados desde la computadora para definir su rutina de operación.

Los microcontroladores tienen limitaciones de velocidad, procesamiento y memoria que los hace poco viables en ciertos contextos. En años recientes se han incorporado microcomputadoras a la producción artística. Tal como los microcontroladores, las microcomputadoras son dispositivos pequeños y fáciles de embeber en estructuras físicas, pero tienen la ventaja de poseer más capacidad de memoria y velocidad y ya cuentan con sistemas operativos tradicionalmente de código abierto. En tiempos recientes se han creado microcomputadoras optimizadas para ciertas tareas tales como audio digital o video digital.

La cultura del código abierto

La cultura del código abierto ha permeado la cultura del arte electrónico. Una extensa cantidad de obras ha sido creada utilizando el sistema operativo Linux en sus diferentes distribuciones. Sin duda, la razón obedece al menor costo y a la posibilidad de utilizar, sin restricciones legales, conjuntos de rutinas encapsuladas y distribuidas en forma de librerías que modularmente permiten diseñar sistemas complejos y robustos sin la necesidad de escribir el código desde cero. La apropiación, la reutilización, la transformación y la adaptación

se vuelven entonces parte intrínseca del entorno creativo y son adoptadas naturalmente en su producción.

La distribución libre de códigos ha permitido que muchas de las iniciativas y propuestas desarrolladas inicialmente por uno, o por pocos individuos, sean adoptadas por una gran comunidad y que esta última aumente y expanda la propuesta medular. El código abierto ha permitido que esfuerzos individuales se vean potencializados por esfuerzos colectivos y unificados.¹²

En la exposición *Pseudomatismos* de Rafael Lozano-Hemmer, en el Museo de Arte Contemporáneo de la UNAM (Urrutia *et. al.*, 2015), el artista realizó una serie de visitas guiadas dirigidas a un público desarrollador en las que explicó detalles técnicos de las obras. Al terminar la visita el artista distribuyó todo el código fuente de las obras. Es interesante analizar el fenómeno, ya que con este gesto el artista establece que, si bien, el código fuente es parte esencial de la obra artística, éste puede ser replicado y modificado en futuros trabajos y en diferentes contextos expandiendo con ello su propia estética y al mismo tiempo resquebrajando parcialmente la noción de autoría.

Aplicaciones para la producción de trabajos interactivos

Será importante citar las herramientas con las que actualmente se está produciendo un grupo significativo de obras electrónicas, sobre todo, dada su adopción dentro de las comunidades artísticas y académicas. En los últimos años algunas de estas herramientas se han convertido en la plataforma de aprendizaje del área tecnológica dentro de los planes de estudio de licenciaturas y posgrados relacionados con la creatividad digital. La carrera en Arte y Comunicación Digitales de la UAM Lerma no es la excepción.

La existencia de software dirigido o adoptado dentro del círculo de producción del arte electrónico no excluye las herramientas genéricas, y en muchas ocasiones la complejidad de las obras requiere códigos finamente diseñados desde la parte más profunda de los sistemas

¹² Un ejemplo claro que me viene a la mente es el lenguaje de programación SuperCollider creado inicialmente por James McCartney pero que, desde que es código libre, es mantenido por una docena de desarrolladores con la constante contribución de un grupo extenso de colaboradores.

digitales, por tanto, es natural el uso de lenguajes de bajo nivel o lenguajes genéricos tales como C y sus variantes, Java y Python, entre otros.

En otras ocasiones la naturaleza de la propuesta es específica de un medio particular y por tal motivo se requiere el uso de un software o dispositivo especializado. Por ejemplo, una obra de bioarte que trabaje con secuencias genéticas requerirá el software utilizado comúnmente en la investigación científica de dicho campo.

La siguiente lista enumera únicamente las herramientas que, como se explicó, han sido adoptadas por la comunidad creativa de manera genérica y que se han adoptado como herramientas de enseñanza del arte digital. Existen muchas otras iniciativas valiosas que no han alcanzado el nivel de popularidad y distribución de las que se mencionan a continuación.

Processing. Creada por Ben Fry y Caes Raes en el Media Laboratory del MIT (2014) es un lenguaje de programación basado en Java optimizado para la manipulación de gráficos en tiempo real. Es continuador del trabajo realizado por el diseñador John Maeda y su propuesta *Design by Numbers* (2001). A diferencia de DBN, Processing adquirió una notoriedad abrumadora dentro de la producción de arte digital y probablemente es el software más utilizado de manera global en la enseñanza de arte digital.¹³

OpenFrameworks. Iniciado por Zachery Lieberman, Theo Watson y Arturo Castro (2016), oF es más bien un conjunto de librerías para C++ que simplifica y unifica diferentes esfuerzos de la comunidad del arte gráfico interactivo. Artistas como Golan Levin, Cory Arcangel y el Graffiti Research Lab, entre muchos otros, utilizan esta infraestructura para la producción de sus materiales recientes.

Arduino. Derivado del proyecto *Wiring* y, basados en el concepto de *open hardware*, Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino y David Mellis (Banzi, 2009) desarrollaron una plataforma para la programación sencilla de microcontroladores. Más allá de su aportación tecnológica, que se limita a la creación de un entorno de simple programación, la propuesta aportó una solución colectiva a una comunidad global de *makers* y *hacedores*. Totalmente asimilada por las nuevas generaciones, esta plataforma es central en la producción de obras interactivas en las que se involucre el cómputo físico.

¹³ Durante los estudios de maestría en el Media Lab de MIT con John Maeda tuve la oportunidad de ver el desarrollo de Processing y utilizarlo en sus etapas iniciales antes de que fuera popularizado por internet.

SuperCollider. Creado inicialmente como un software comercial por James McCartney en el año de 1996, el código fuente fue liberado como código libre en 2002 y a partir de ese momento una gran cantidad de artistas sonoros y compositores lo ha adoptado para la producción de música interactiva y algorítmica (Wilson, Cottle y Collins, 2011). Con el tiempo, SuperCollider se posicionó como el heredero de los lenguajes de programación de audio históricos, tales como Music y cSound.

PureData y MaxMSP. Desarrollado por Miller Puckette desde 1988, PureData y su contraparte comercial MaxMSP merecen particular atención por ser lenguajes gráficos con curvas de aprendizaje sencillas y son, por tanto, excelentes opciones para introducir a los estudiantes en la programación computacional (Winkler, 1998). Al mismo tiempo ofrecen técnicas simples para la implementación de obras interactivas.

Sensores

A lo largo de este capítulo se ha hablado de cómo las actividades de los espectadores y del público son utilizadas como mecanismos que se incorporan en las obras interactivas. ¿Con cuáles mecanismos se cuenta para identificar la actividad del público? ¿Cómo podemos registrar la información, datos y acciones del público participativo? Centrados en el campo digital es común hablar de sensores como técnica de medición de información. La información de estos sensores es normalmente analógica, por lo que dichos datos deben pasar por un proceso de conversión analógico-digital que permite, al final de la cadena de registro, tener valores binarios con cierta resolución y velocidad de muestreo.¹⁴

Existe una diversidad de sensores digitales pues son el punto de partida para una extensa gama de prácticas tecnológicas, industriales, científicas y de entretenimiento. A continuación se enumeran algunos de los diferentes tipos de sensores que se utilizan en el campo del arte digital interactivo.

Cámaras y micrófonos. Estos dispositivos en sus diferentes variantes nos permiten extraer gran cantidad de información del entorno. Requieren necesariamente un análisis de los

¹⁴ Los aspectos relacionados con el procesamiento de señales digitales son importantes en el campo del arte digital por lo que se recomienda un estudio profundo del tema. Un buen punto de partida es el libro *Procesamiento de señal* de Steven W. Smith (1999).

datos obtenidos, pero con adecuados procesamientos de señal ofrecen información precisa y de alta resolución. Con una cámara y utilizando sistemas de visión por computadora es posible inferir, por ejemplo, la cantidad de personas en un espacio determinado o el estado emocional de una persona. Un micrófono y un sistema de escucha automática nos pueden ofrecer información relacionada con un entorno y las situaciones que en él acontecen.

Sensores de movimiento y presencia. Son dispositivos que permiten medir y cuantificar el movimiento de un cuerpo. Entre los dispositivos básicos encontramos los sensores infrarrojos, los sensores PIR, los sensores ultrasónicos, etc. Evidentemente, las cámaras de video son también dispositivos que, después de un proceso de análisis, permiten la identificación de movimientos. En estas líneas equipos como el Kinect y el LEAP Motion son finalmente cámaras infrarrojas con un preprocesamiento interno de filtraje, análisis y clasificación.

Sensores de presión y fuerza. Dependiendo de su configuración, estos sensores pueden ser utilizados para medir peso, flexión y torque. También pueden ser utilizados para registrar, por ejemplo, la fuerza de un soplo o la inhalación y la exhalación de un individuo.

Medición de interfaces de cómputo. Para las obras en red y computadora es posible, evidentemente, medir el uso de los dispositivos tradicionales de una computadora, como son el teclado, el ratón, los dispositivos táctiles, etc. En la actualidad es posible realizar diversos y sofisticados análisis del uso de estos sensores de entrada, lo que permite inferir conductas complejas de los usuarios y también las relaciones entre varios usuarios.

Sensores de entorno. Llamamos así a los sensores que nos permiten hacer mediciones de nuestro entorno. En el nivel básico tenemos sensores de luminosidad, humedad, acelerómetros, giroscopios, entre otros. Aquí también podemos hablar del uso de cámaras que después de un proceso de análisis gráfico pueden derivar flujos y direcciones de partículas, fluidos, personas, tráfico, etcétera.

Sensores biométricos. En los últimos años se han desarrollado también sensores de medición de eventos fisiológicos en humanos tales como dispositivos para medición de ondas cerebrales, cardiovasculares, musculares, etc. La colección de obras va en aumento y

podemos ver una extensa producción de obras preocupadas por utilizar señales de ondas cerebrales con fines artísticos.

Mapping

Uno de los temas fundamentales dentro del arte que contiene propiedades interactivas está relacionado con lo que en inglés se conoce como *mapping* (mapeo en español) y que tiene que ver con las reglas y relaciones que se estipulan entre los valores de entrada proporcionados por los participantes, comúnmente medidos por los sensores, y su impacto en la obra artística resultante. La manera en la que se decide cómo afecta un determinado valor de entrada a una o varias variables dentro del sistema de la obra desempeña un papel vital dentro de la producción del arte electrónico. En situaciones multimedia, donde se juega con una diversidad de técnicas funcionando al mismo tiempo, un parámetro de entrada puede afectar a uno o varios elementos de salida de manera indistinta a selección del creador. ¿Cómo se decide cuál valor de entrada impacta a cuál parámetro de una obra dinámica? La respuesta está dada fundamentalmente en el contexto y el concepto de la obra misma; sin embargo, el tema es tan delicado que justo aquí es donde radica la maestría y el dominio del medio. La intuición y la destreza son fundamentales en este punto del proceso creativo de manera equivalente a las decisiones que toma un compositor a la hora de hacer una orquestación.¹⁵

En el caso del arte multimedia la libertad de selecciones y combinatoria es aún más extensa y compleja, ya que los valores de entrada de un medio particular pueden traducirse e interpretarse en otro medio diferente del original. Un ejemplo lo encontramos en mi obra *Delta 16.57/-99.00* (Benítez, 2015) en la cual los datos meteorológicos son utilizados para generar estructuras sonoras.

Mapeo lineal y mapeo complejo. En el mapeo lineal existe una relación directa entre un parámetro de entrada y su impacto en la obra. Por ejemplo, la obra *Tensión superficial* (Urrutia et. al., 2015) de Rafael Lozano-Hemmer mide la posición angular del espectador en relación con la pantalla de proyección, y en función de dicha posición selecciona una de las varias imágenes de un mismo ojo mirando hacia posiciones diferentes. Hay una relación

¹⁵ Mi tesis de maestría (Solís, 2004) contiene un ejemplo de mapeos audiovisuales y gestuales.

directa entre la posición del espectador y la imagen seleccionada. De esta manera se genera la sensación de que el ojo sigue al espectador libremente. En este ejemplo se puede observar que el mapeo seleccionado relaciona la posición del espectador con una imagen particular dentro de un banco de imágenes.

En los mapeos complejos, por el contrario, las relaciones entre la entrada de valores y su impacto dentro de la obra o no son de tipo uno-a-uno, o sus relaciones entrada-salida cambian de manera compleja en función de otros parámetros. Los mapeos complejos son más complicados de modelar, pero en muchas ocasiones pueden ofrecer mecanismos de expresividad más cercanos a nuestra percepción. Son utilizados cuando se busca la producción de obras menos predecibles en donde la pieza busca emular actividades y conductas complejas. Myron Kruger nos dice, en relación a su obra *Responsive Environments* de 1977: “Las relaciones hombre-máquina se limitan usualmente a un hombre *picando* en una máquina con sus dedos o tal vez ondeando una varita sobre una tableta. Hace siete años estaba insatisfecho con las restricciones de este tipo de diálogos y me embarqué en una investigación que exploraba maneras más interesantes para que los hombres y las máquinas se relacionaran” (Abad, 2006).

Más recientemente, en el año 2000, Ken Rinaldo creó, por ejemplo, una comunidad de esculturas con sonidos robóticos que establecen un comportamiento colectivo emergente en donde, sin reglas de coordinación centralizadas se logran relaciones complejas a partir de estructuras simples pero repetidas de coordinación (Shanken, 2013).¹⁶

El tema de la complejidad en los mapeos y las relaciones entre información producida por los espectadores y su resultado dentro de la obra artística nos obliga a hablar sobre la programación, el modelado y la algoritmia en el arte interactivo.

Programación, modelado, emulación y algoritmia

Una de las plataformas del arte electrónico con componentes interactivos es digital tanto en su producción como en su manifestación. Una de las virtudes principales de los sistemas digitales es la simplificación de los procesos de transformación de materiales. Dichas transformaciones pueden ejecutarse a una velocidad tal que muchas de ellas entran dentro de las

¹⁶ Es importante mencionar que mucho de mi trabajo se enmarca dentro de estas lógicas creativas, así como en la búsqueda de mecanismos de interacción compleja.

conocidas transformaciones en tiempo real, es decir, aquellas que se ejecutan con tal rapidez que para la percepción humana son inmediatas. Estas manipulaciones son producto de funciones escritas y desarrolladas en lenguajes de programación. Independientemente de si fueron creadas por el equipo de desarrollo de un *software* cerrado, o si fueron desarrolladas *ex profeso* para una obra artística (por el creador o su grupo de trabajo), su implementación se llevó a cabo por una mente humana que fue capaz de hacer la traducción entre el problema real y su solución lógico-numérica dentro de una gramática de programación computacional.¹⁷

La programación computacional es uno de los eslabones no visibles dentro del discurso del arte interactivo, ya que los detalles en la implementación de una solución impactarán en mayor o menor medida la funcionalidad de la interactividad en la obra artística, de manera análoga al modo en que, en el lenguaje natural, la selección de uno u otro adjetivo maticen y delimiten los detalles de una idea verbal.

La programación computacional es, por tanto, una especie de máquina traductora entre los conceptos y pensamientos artísticos y su realidad material. En el arte interactivo digital las relaciones entre los valores de entrada y su impacto en el discurso artístico se logran justamente a través de funciones descritas mediante una programación computacional y, por tanto, es de suma importancia tenerlas presentes al menos durante el análisis técnico de una obra.

Muchas de las obras interactivas buscan generar discursos en donde los materiales responden y generan patrones coherentes y lógicos; patrones que responden de manera *orgánica* a los estímulos de los espectadores. El concepto de orgánico es difuso, pero nos permite discernir entre materiales ilógicos dentro de un contexto determinado y otros que coinciden y se integran con el discurso artístico. Lograr interactividad orgánica y coherente requiere la creación de modelos y emulaciones dentro del lenguaje de programación.

¹⁷ En relación con la programación y el arte es importante hablar de la figura del artista programador y del artista desarrollador que se ocupa no sólo de gestar la idea creativa, sino también de realizar su implementación. Se comenta este dato porque existe gran cantidad de artistas electrónicos que no son los desarrolladores de las soluciones técnicas. En mi opinión es fundamental que la nueva generación de artistas digitales tenga un conocimiento profundo de las herramientas con las cuales trabaja.

Un modelo es una abstracción que, simplificando la realidad, permite generar una representación de un fenómeno, evento o estructura y mantenerla aislada y controlada. Para la generación de un modelo es necesario hacer una serie de algoritmos que en su conjunto funcionan de manera similar al objeto real, pero que permiten manipular partes del evento y hacer variaciones controladas. El modelado es, por tanto, un elemento importante de las obras interactivas complejas. La complejidad en el siglo XXI puede y está llegando a niveles extremadamente sofisticados de análisis y de control que permiten realizar, por ejemplo, ciertos tipos de inferencia, predicción y automatismo inteligente. En este sentido conceptos como *big data* e inteligencia artificial se integran como exploraciones importantes en el arte electrónico.

Inteligencia artificial, Machine Learning Big Data y Deep Learning

Como se vio anteriormente, una de las intenciones de algunas obras de arte electrónico interactivo es producir diálogos complejos entre los espectadores y su resultado dentro de la obra artística. Para algunos artistas y críticos las fórmulas lineales son limitadas, poco expresivas y no generan posibilidades de lecturas múltiples de las obras. La inclusión de técnicas de análisis, clasificación y predicción modernas son importantes para lograr llevar los conceptos de interactividad en el arte a niveles de mayor complejidad y sofisticación en las relaciones espectador-obra.

Los términos inteligencia artificial, *Machine Learning Big Data* y *Deep Learning* son definiciones que se utilizan en las ciencias de la computación para hablar de conjuntos de rutinas, algoritmos y procedimientos que permiten la organización, clasificación y análisis de información. El término más genérico de inteligencia artificial nos establece los principios básicos estadísticos sobre los que se fundamentan estas herramientas.

El *Machine Learning* o aprendizaje de máquina se enfoca a diseñar metodologías que permiten a un sistema digital inferir y clasificar nuevos materiales en función de una colección previa de materiales similares. Aplicados correctamente son mecanismos que permiten

emular respuestas de salida coherentes a los ejemplos de entrada y, por tanto, son ideales para emular y simular respuestas humanas.¹⁸

La *Big Data* se encarga del estudio de datos masivos y de la búsqueda de patrones dentro de dichas fuentes de numerosos elementos. Se suele asociar con el trabajo en la nube, con el uso de supercomputadoras y el cómputo paralelo debido a la imposibilidad de las computadoras tradicionales para trabajar con dicha cantidad de información.

Finalmente, el *Deep Learning* o aprendizaje profundo es un área del aprendizaje de máquina que se encarga del modelado de estructuras complejas a través de lo que se conoce como redes profundas. Se ha detectado una eficiencia notoria en algunos campos como la representación musical y, por tanto, impacta la potencialidad de la representación artística.

El arte digital y los nuevos medios están fuertemente vinculados con la cultura digital y las tecnologías de información y comunicación. Si estas tecnologías se actualizan constantemente y adoptan las herramientas que los teóricos e investigadores generan, es natural que dichas innovaciones impacten la producción artística digital. La manera más directa, pero menos evidente, se encuentra en las herramientas que los artistas utilizan para producir sus obras. Sin embargo, la manera más profunda es a través de obras que directamente utilizan las nuevas técnicas para proponer materiales que difícilmente se habrían obtenido sin dichos avances tecnológicos.¹⁹

CONCLUSIONES

Como se puede apreciar a lo largo de este capítulo, la interactividad dentro del campo del arte electrónico es fascinante. Es un campo con cierta madurez conceptual pero lo suficientemente joven como para delinear sus fronteras. Sus repercusiones en el mundo del arte aún no son totalmente claras, pero sus aportaciones dentro del campo del arte electrónico son contundentes. Al estar estrechamente ligado a los nuevos paradigmas sociales en donde

¹⁸ En el campo del sonido el autor ha trabajado en el área del *Machine Listening* con la finalidad de crear sistemas de predicción de entornos acústicos (Solís, 2006).

¹⁹ Es interesante observar la reciente creación de Kadenze, un mooc especialmente enfocado a la tecnología en el arte. Uno de los cursos que al día de hoy se está ofertando es el de *Machine Learning for Musicians and Artists* (Kadenze, 2017) basado en el trabajo de Rebecca Fiebrink (2011).

las tecnologías de la información juegan –cada día– un papel más preponderante, el arte interactivo electrónico tiene implicaciones cada día más directas dentro de la sociedad.

La cantidad de artistas, espacios de exhibición, instituciones de arte y académicas y, sobre todo, espacios públicos cautivos va en aumento y es muy probable que las nuevas generaciones de artistas produzcan obras cuyos conceptos, metodologías y técnicas transporten la noción de interactividad a niveles de complejidad que desfiguren aún más los conceptos mismos de arte, creatividad y expresión.

En el terreno tecnológico, los avances cotidianos en las tecnologías de la información, la creación y masificación de sensores y sistemas de detección, la evolución en las técnicas de análisis y de clasificación que permiten el modelado de sistemas complejos y la predicción de conductas humanas, producirán en el mediano plazo, obras de corte interactivo que cuestionen las nociones de inteligencia y humanización. ■■■



Música y escena

En este capítulo se presentan los trabajos de música y escena, todos ellos creados con desarrollo tecnológico propio. Muchos de éstos fueron desarrollados utilizando el software Juum que se describe en el capítulo 17. Algunos son proyectos solistas, otros son colaboraciones grupales y otros más involucran el trabajo con escenógrafos y bailarines.

TRÍO NICROM

Fecha de producción: 2001. *Lugares de exhibición:* teatros del Instituto Mexicano del Seguro Social de la Ciudad de México. *Material:* Conciertos para electrónica, teclado GEM, saxofón y laptop. *Tecnologías creativas:* Java, JSyn.



Figura 3.1. | Presentación del Trío Nicrom.

El trío de improvisación de música electroacústica Nicrom surgió a mediados del año 2000 como respuesta a la inquietud de crear un espacio de creación colectiva en donde la música y la pintura tuvieran un valor de igual importancia. De esta manera, la propuesta del trío era que el aspecto visual fuera un factor fundamental para la parte auditiva y de la misma forma se buscaba considerar al material sonoro como parte necesaria para el trabajo visual. El trío mantenía un especial interés en explorar y explotar las posibilidades y los recursos de nuevas tecnologías electrónicas, ya fuera procesando en tiempo real instrumentos acústicos, o por medio de sintetizadores y computadoras. La intención del trío era encontrar mecanismos alternos para producir discursos musicales coherentes basados más en la manipulación tímbrica y gráfica que en las relaciones interválicas de los sonidos, lo cual llevó al trío a trabajar y explorar el espectro del ruido y de la pintura abstracta.

Las presentaciones del trío Nicrom fueron en su totalidad música improvisada durante las funciones en las que se proyectaba en el escenario el material gráfico que el artista visual creaba en tiempo real durante todo el concierto. La forma musical, el material tímbrico y gráfico, así como las relaciones jerárquicas entre los diferentes miembros del trío se iban definiendo y modificando constantemente durante las ejecuciones. La libertad que ofrece esta técnica de creación colectiva, que en un inicio podría parecer un proceso sin control y sin dirección, obligaban por el contrario a todos los miembros del ensamble a mantener la atención no sólo en su propio material sino también en el material y las propuestas que el resto de los integrantes proponía a lo largo del concierto.



Figura 3.2. | Presentación del Trío Nicrom en los teatros del IMSS.

De enero a mayo de 2001 el trío Nicrom realizó una temporada de conciertos que le permitió explorar y experimentar con diferentes propuestas musicales, variados recursos tecnológicos y, sobre todo, con diferentes procesos creativos. Cada concierto fue diferente y en cada uno de ellos se obtuvieron resultados que permitían mejorar y madurar la propuesta musical. En algunas de las presentaciones se contó con la participación de músicos invitados que propiciaban la renovación de ideas y sonoridades musicales.¹

IMPROVISACIONES PARA PIANO Y MEDIOS ELECTRÓNICOS

Veo cada una de mis improvisaciones como una instantánea del estado en el desarrollo de mi investigación acústica/técnica/creativa. El estilo, las tecnologías y las intenciones expresivas han cambiado significativamente a través de los años, pero el uso del piano acústico se ha mantenido como una constante. Incluso cuando improviso sólo con aparatos electrónicos o visuales trato el material con estrategias similares donde la noción de *estructura* –a veces reconocible, a veces sugerida– es de extrema importancia.



Figura 3.3. | Experimental Festival-Sound, Art & Performance. ©Tokyo Wonder Site, Fotografía de Toshiya Suda. Improvisación para piano con motores dc en el Tokyo.

¹ El trío estaba integrado por Rodrigo Garibay en el saxofón, Hugo Solís en los teclados y el software jsync y jmsl, y Mauricio Zárate en los visuales. Entre los músicos invitados estuvieron Marcia Medrano en el trombón y Alex Otaola en la guitarra entre otros.

Otro elemento importante en mi trabajo es el potencial expresivo de la interacción hombre-computadora. Este interés me ha llevado a crear agentes digitales que no tratan de reemplazar a los músicos en vivo, ni de sólo ampliar acústicamente al instrumento, sino que más bien proponen paradigmas de composición donde las máquinas puedan enfatizar la expresión humana.² Con este pensamiento he realizado gran cantidad de conciertos. En muchos de ellos he llevado a cabo exploraciones en otros campos buscando la integración de datos y parámetros de diferentes contextos. Por ejemplo, en una ocasión, utilicé una chaqueta desarrollada por Maxel Keene que detectaba mis movimientos y añadía una nueva capa de análisis al componente sonoro digital.

ENSAMBLE CRUMBLE

El Ensamble Crumble fue un conjunto de improvisación electroacústica formado a principios de 2005. La mayoría de los miembros pertenecían al Grupo de Tecnología Musical de la Universidad Pompeu Fabra y la mayoría de nosotros teníamos estudios interdisciplinarios en música y tecnologías del sonido. El ensamble fue creado como una oportunidad para desarrollar ideas musicales en forma de improvisaciones electroacústicas libres. El ensamble realizó una vasta colección de grabaciones además de tres conciertos en diferentes lugares de Barcelona.³

El deseo inicial era que cada miembro pudiera fusionar su instrumento con tecnologías digitales. Sin embargo, después de algunos meses el grupo tomó un camino acústico y sólo el autor siguió trabajando con dispositivos electrónicos. Durante el año y medio que el grupo estuvo activo dos integrantes abandonaron el ensamble y dos nuevos miembros se integraron en diferentes momentos. No obstante, todas las grabaciones que se seleccionaron para el análisis se realizaron con los mismos miembros.

² La improvisación con piano y medios electrónicos la realicé de manera constante hasta el año 2010. En ese momento dejé el piano, sin embargo, en tiempos recientes he retomado la actividad. Un último concierto se llevó a cabo en abril de 2018 en la Casa del Tiempo de la UAM.

³ El trabajo del ensamble fue utilizado como material para mi tesis de maestría (Solís, 2006) en la cual se realizaron procedimientos computacionales con el objetivo de identificar patrones musicalmente coherentes y relaciones lógicas en el contexto de la música experimental colectiva.

Los procedimientos de trabajo del ensamble eran los tradicionales con métodos comunes de ensayo. Nos reuníamos una o dos veces por semana y tocábamos durante dos o tres horas. Todos los ensayos tenían lugar en el estudio de audio IUA-Phonos. Se realizaban improvisaciones muy largas de aproximadamente 40 minutos cada una en cada sesión. Después de una improvisación se realizaba una discusión informal donde cada integrante explicaba su propia impresión del resultado.

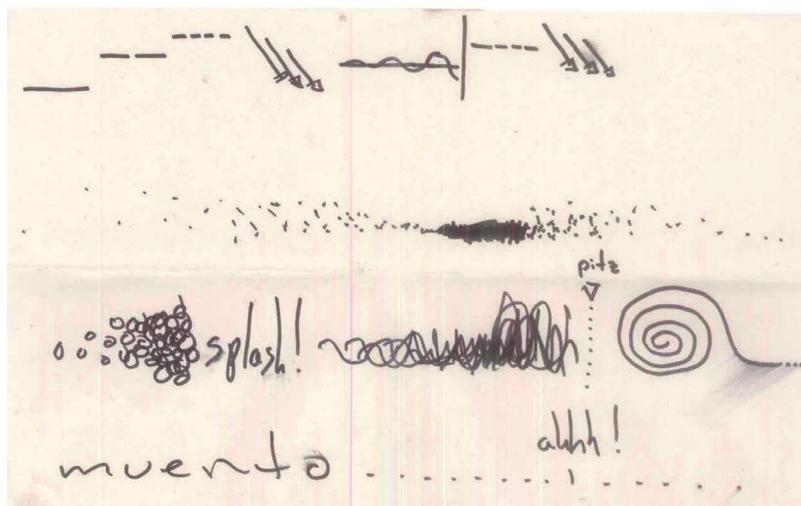


Figura 3.4. | Partitura gráfica para estructura de improvisación controlada del Ensamble Crumble en la galería Metronom en Barcelona.

Uno de los mayores problemas que el conjunto enfrentaba era la tendencia de fusionar secciones de música que no tenían relación entre sí. Varias veces los miembros pensaban que las improvisaciones perdían dirección y que los contenidos se hacían redundantes. Por esta razón en algunos ensayos se desarrollaron ejercicios específicos con la intención de resolver esa falta de dirección. La mayoría de estos ejercicios impone cierta restricción a la improvisación como “improvisa todo el mundo en *metzopiano*”, “evita tocar si más de dos ejecutantes están improvisando en ese momento”, “improvisa respondiendo al material de otro ejecutante únicamente”.

El Ensamble Crumble tenía un lenguaje de improvisaciones estructuradas, lo que se traducía en buscar desarrollar ideas musicales que tuvieran exposición, desarrollo y conclusión.

Para alcanzar este objetivo se realizaban variados experimentos y mecánicas de ejecución. Uno de los ejercicios era la musicalización de historias y eventos de dos minutos. Por ejemplo, se establecía que musicalizaríamos o representaríamos en dos minutos la caída de un vaso de vidrio y su rompimiento en múltiples fragmentos. O por ejemplo, improvisar sobre la imagen del movimiento interior de un río y cómo el río mueve objetos flotantes en la superficie.

Miembros del ensamble

Con el fin de obtener una mejor idea sobre el conjunto se presenta una breve descripción de los instrumentos y los miembros con los que se hizo el trabajo del análisis computacional.

Maarten Grachten (Países Bajos, 1976) tocaba la guitarra eléctrica. Maarten Grachten comenzó a tocar la guitarra a la edad de 10 años y tomó lecciones de guitarra clásica y eléctrica durante varios años. De 1989 a 2001 tocó guitarra y batería en varios grupos de rock experimentales, a menudo con un enfoque basado en la improvisación. Dentro del ensamble el instrumento nunca fue procesado electrónicamente y la mayor parte del tiempo tocó de una manera “tradicional”. Sin embargo, a veces las técnicas extendidas que enriquecen la paleta de sonido eran observables: los *glissandi* y el movimiento de las clavijas de afinación. En ocasiones se empleó el rasguño de las cuerdas con clavos y objetos externos.

Sylvain Le Groux (Francia, 1977) tocaba también una guitarra eléctrica. Sylvain Le Groux estudió flauta clásica y teoría de la música en el Conservatorio Nacional de Nantes, armonía básica y contrapunto en el Conservatorio de París y música electroacústica en Studio Delta P (La Rochelle, fr.). Dentro del ensamble, Sylvain tocaba con pocos efectos del amplificador y la mayor parte del tiempo lo hacía con técnicas que producían ruidos y sonidos fragmentados. La barra de vibrato era ampliamente utilizada y también el uso de objetos de metal para percutir las cuerdas.

Alfonso Pérez (España, 1977) tocaba el violín acústico. Alfonso Pérez estudió violín en el Conservatorio de Palencia y Ciencias de la Computación en la Universidad de Valladolid (España). Su formación en música es principalmente con música clásica y folk. Alfonso había tocado con varias orquestas y conjuntos musicales en España y Austria. Dentro del ensamble, Alfonso tocaba la mayor parte del tiempo de una manera estándar. Sin embargo,

a veces se empleaban glissandi y armónicos naturales. El instrumento se amplificó con una ganancia variable que permite capturar sonidos muy delicados. En algunos momentos el cuerpo del instrumento fue incluso utilizado como resonador de percusión con los dedos.

Gabriela Villa (Ciudad de México, 1976) tocaba la viola. Gabriela Villa había colaborado en muchos proyectos interdisciplinarios y había sido miembro de varias orquestas de México. Estudió viola y análisis musical en el Conservatorio de las Rosas de Morelia, México. Realizó sus estudios musicales básicos en la Universidad Nacional Autónoma de México, en la Ciudad de México. Dentro del ensamble Gabriela interpretaba con técnicas extendidas, armónicos naturales y artificiales los cuales eran ampliamente utilizados. También se usaron *Scordaturas*, *sul tasto*, *sul ponticello* y *Col legno*. El instrumento también era amplificado.

Por mi parte yo tocaba un sintetizador Kurzweil κ2000 controlado con un teclado M-Audio Oxone. El banco de sonido durante todas las improvisaciones se limitó a 10 sonidos, sin embargo, el timbre de cada sonido podría ser alterado drásticamente con los mandos del controlador. Todos los sonidos compartían una calidad ruidosa y algunos podían producir patrones semi-rítmicos que eran empleados moderadamente. Ninguno de los sonidos tenía una envoltura larga que pudiera generar texturas densas sin un esfuerzo constante en la ejecución.

DÚO JUUM

El Dúo Juum (<http://duojuum.net>) está integrado por Gabriela Villa quien toca la viola y recita poemas de su autoría mientras yo toco el piano y controladores diversos. Es un dúo experimental electroacústico y surgió como un espacio para la exploración de nuevas sonoridades y nuevos métodos de interacción entre ambos ejecutantes y los medios digitales. El nombre Juum, que en maya significa “ruido”, da la dirección estética al dúo que entre 2006 y 2008 tocó en Italia, México, Gran Bretaña, Estados Unidos y Japón. Se fundó en Barcelona, España, como un derivado del Ensamble Crumble. Ahí el dúo produjo el disco *La nave de los locos* pensado como una travesía por los caminos subjetivos de los viajes oníricos.



Figura 3.5. | Portada de *La nave de los locos*, primer material del Dúo Juum.

En 2007 produjo *Vértice* como una experiencia audiovisual. En 2008 produjo *Conjuro de la memoria* basada en un poema original extendida con pianos muertos.⁴

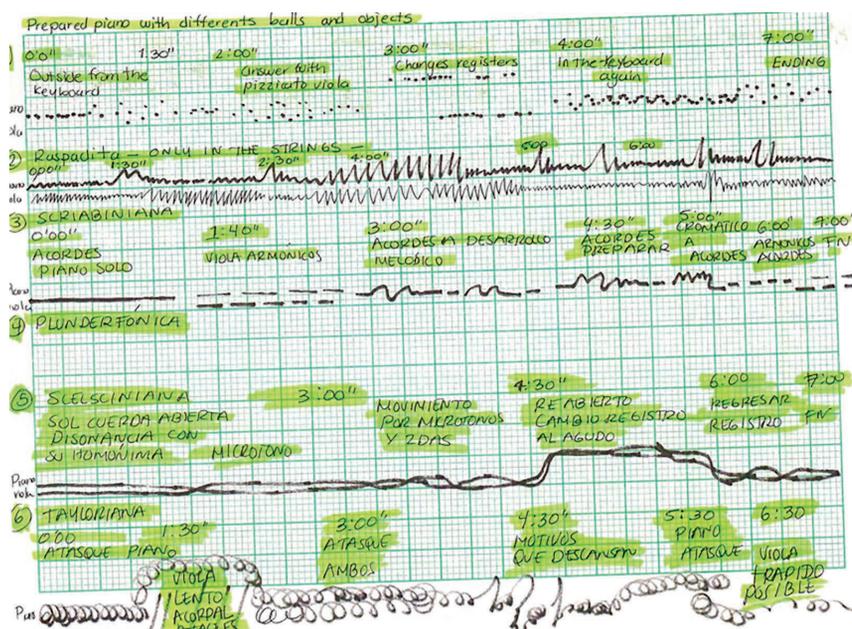


Figura 3.6. | Partitura gráfica de una obra del Dúo Juum.

⁴ El concepto de los pianos muertos será recurrente en mi trabajo y hace referencia al uso en instalaciones o a la ejecución de almas de pianos a los cuales se les removió el teclado y el mecanismo dejando libre acceso a las cuerdas para ser percutidas y frotadas directamente.

Un programa de mano escrito por Gabriela Villa hacía referencia a la lógica del dúo:

Juum significa ruido en maya y responde a la inquietud de experimentación sonora del dueto. El dúo Juum, integrado por Hugo Solís y Gabriela Villa, es un ensamble de improvisación electroacústica libre creado a principios del año 2006. Ambos músicos, originarios de México, tienen intereses y estudios interdisciplinarios que incluyen la música, la tecnología musical, el medio digital y la palabra y, por ello, el dúo Juum ofrece respuestas a la intención de desarrollar ideas a partir de la improvisación musical y del medio digital con la determinación de convocar a un escucha que quiera desautomatizarse a través de la palabra, combinando la expresividad de una voz acústica (poesía) con tecnologías de la maleabilidad del medio digital que dan como resultado una sonoridad alterna manipulada en tiempo real.⁵

Desde un aparato estético desprovisto de las reglas del momento y alejado de la música de los medios masivos de comunicación —el acto de improvisar libremente a través de una especie de forcejeo con el aparato estético mismo—, el dúo Juum ofrece la posibilidad de traspasar los límites manteniéndose en una frontera que abre la ventana a la realidad de sus dos integrantes.

En la música del dúo Juum conviven espectros de épocas distantes en un espacio construido por sonoridades en las que el contenido energético de la acción que las hace reales es de gran importancia. La contemplación de la naturaleza del sonido puro e impuro, la disolución del tiempo y su aleteo formal nos ubican frente a nuevos parámetros musicales donde la realidad de la materia misma se vuelve eje del acto discursivo. Resurgiendo desde una voz acústica (viola) en combinación con los sonidos creados vía electrónica, encontramos en el dúo Juum fragmentos puestos en relación con elementos que antes les eran ajenos: corrientes de aire surgidas del caos.

El escucha que dentro de la música busca un asidero que le dé seguridad, se verá confrontado durante la experiencia sonora de la improvisación libre, una construcción que lo invitará y lo situará frente a la acción de la escucha activa. La música puede haber muerto, pero sólo para devenir energías sonoras liberadas, experiencias sensoriales radicales que no ceden ante lo decorativo ni ante el hedonismo, un material que intentará sorprender los instantes siguientes.

MÚSICA PARA DANZA

La producción de música para danza fue una constante durante mucho tiempo. De 1997 al 2000 improvisé todos los lunes por la tarde para el grupo de danza experimental de la coreógrafa Mirta Blonstein. Cada año se hacía una presentación pública en la Caja Negra de la escuela de danza del Centro Nacional de las Artes en la CDMX.

⁵ El Dúo Juum se mantuvo inactivo de 2008 a 2017; no obstante, en tiempos recientes ha retomado su actividad. En diciembre de 2017 tocó en la Casa del Lago en la CDMX y, en agosto de 2018, en Daegu, Korea.



Figura 3.7. | Piano en vivo para la coreografía del grupo Viandante.

Posteriormente, en 2008, participé en el proyecto *3DDance concert*. Esta colaboración artística exploró las dualidades reales y metafóricas entre los escenarios virtuales físicos y proyectados gráficamente, así como los sonidos acústicos y electrónicos. Siete espacios confinados eran explorados tanto por una bailarina como por el músico que, junto con el piano acústico (ampliado con dispositivos electrónicos), coexistían en ambos dominios.



Figura 3.8. | Momento de diálogo entre pianista y bailarina.

En este trabajo los colaboradores combinábamos nuestros conocimientos y habilidades para crear una experiencia multimodal híbrida compuesta por un bailarín humano ampliado digitalmente con gráficos dinámicos generados a partir del movimiento de la bailarina y registrados con una cámara; y un piano acústico ampliado digitalmente con sonidos manipulados por computadora que generaba una textura electrónica en función de los materiales generados en el piano. Además, yo llevaba puesto un chaleco con acelerómetros que servía como generador de parámetros complementarios a los datos acústicos que recibía la computadora.

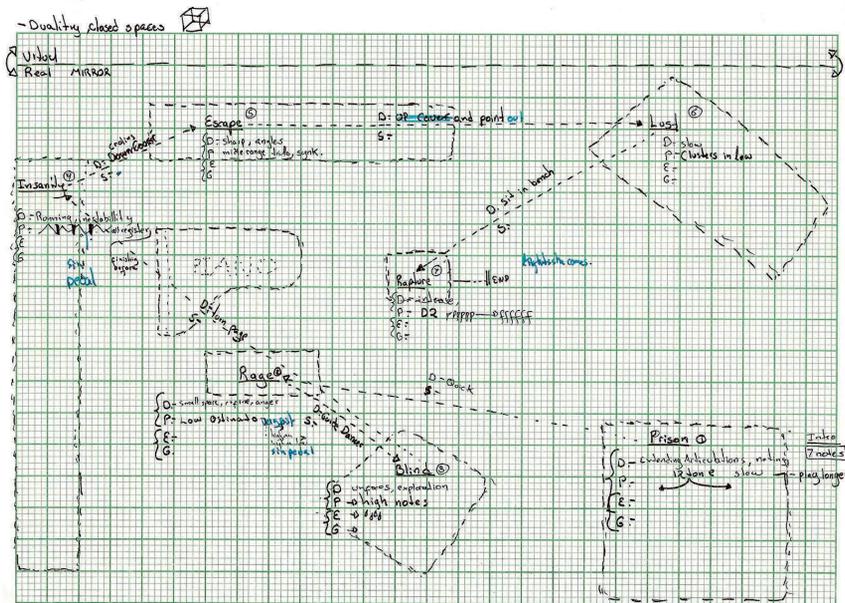


Figura 3.9. | Partitura gráfica para la pieza *Just a Minute*.

Esta obra fue parte de un ciclo de conciertos que fomentaba el diálogo entre la danza y el arte digital. La ficha técnica de esta obra es la siguiente:

Título: Just a minute. *Coreografía:* Maria Cynthia Anderson y Devin McDermott. *Sonido y composición musical:* Hugo Solís. *Gráficos generativos:* Hugo Solís. *Vestuario:* Maria Cynthia Anderson y Devin McDermott. *Bailarina:* Devin McDermott. *Pianista:* Hugo Solís. ■■■



Radiobanda

Hoja de sala

Fecha de producción: 2002. *Lugares de exhibición:* Media Laboratory, MIT, Cambridge USA.

Material: Radios, servidor web. *Tecnologías creativas:* MaxMsp, Processing.

DESCRIPCIÓN¹

RadiobANDA es un proyecto multimedia que tiene como finalidad que una estación de radio recree, intervenga y transforme el total del material transmitido en todo el ancho de banda en donde dicha estación emite su señal. Esta intervención se logra con la directa participación de los radioescuchas. Para ello una dirección de internet contiene un gráfico interactivo que al ser modificado genera las transformaciones de varias estaciones de radio. Las diferentes señales de las emisoras son producidas por una colección de radios analógicos que en su conjunto

¹ **banda**¹ (Del fr. ant. *bande*, bende, y este del franco *binda*, lazo, cinta). de frecuencia. 1. f. *Telec.* En radiodifusión y televisión, intervalo de frecuencias entre dos límites definidos, que condiciona su aplicación.

banda² (Quizá del gót. *bandwo*, signo, bandera). 4. f. Pandilla juvenil con tendencia al comportamiento agresivo. 7. f. Conjunto de instrumentistas, con o sin cantantes, que interpreta alguna forma de música popular. **a dos**, o **tres**, etc., **s. 1.** locs. adjs. Dicho del billar: Que se juega de modo que la bola impulsada deba tocar varias **bandas** de la mesa antes de hacer carambola. U. t. c. locs. advs. *Jugar a tres bandas.*

forman una instalación. Estas señales son enviadas a un servidor de internet controlado por el gráfico interactivo.

Conceptualmente el proyecto tiene varias finalidades: hacer que los escuchas hagan suyo el espectro total de la radio, crear un puente acústico entre los diferentes radioescuchas y ser un mecanismo para que la radio escuche a su público. Al mismo tiempo se busca que cada uno de los elementos que conforman el proyecto sea por sí solo un trabajo estético que hable directa o indirectamente sobre la radio.

FUNCIONAMIENTO

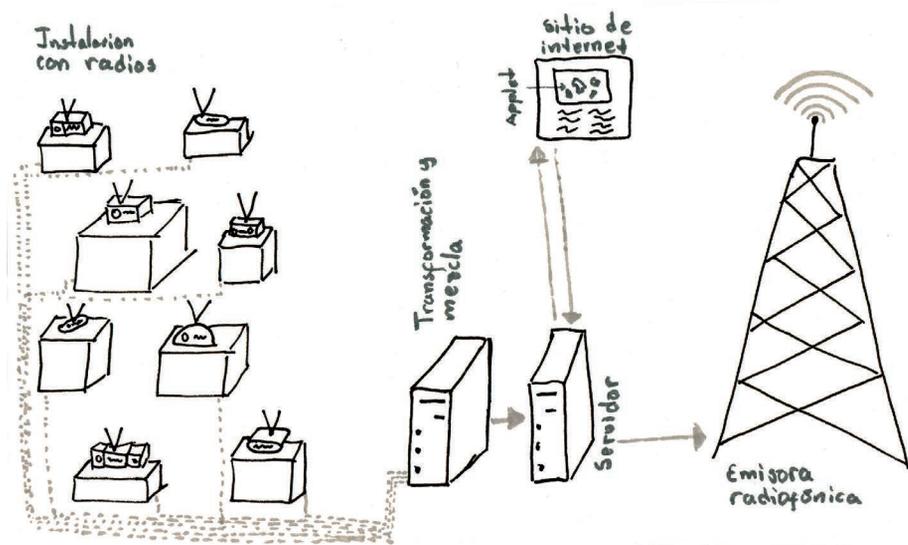


Figura 4.1. | Esquema de funcionamiento de radiOBANDA.

RadiOBANDA está conformado por varios elementos: una instalación de radios analógicos; un servidor de internet que contiene la página que los usuarios utilizan para modificar el material producido por la instalación y la estación de radio encargada de transmitir el material que los radioescuchas modifican.

La instalación de radios es una colección de radios de diferentes características, cada uno de ellos sintonizado en una estación diferente buscando abarcar todo el rango de fre-

cuencias. Idealmente, la instalación deberá estar montada en alguna sede relacionada con la radio o en algún lugar público. El lobby de la estación radiofónica o alguna Casa de Cultura son lugares tentativos. El material acústico de todos los radios es enviado a una computadora que se encarga de hacer las transformaciones sonoras de acuerdo con la información recibida por el gráfico interactivo.

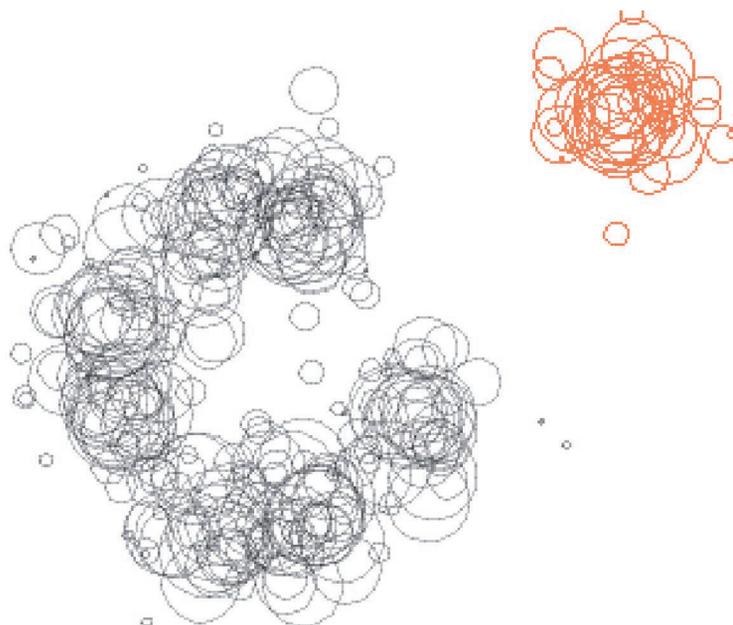


Figura 4.2. | Gráficos interactivos de radiOBANDA.

En la dirección www.radioBANDA.net el radioescucha puede acceder al gráfico que controla los parámetros de transformación de los radios. El gráfico interactivo consta de varios objetos. Cada uno de estos objetos representa uno de los radios de la instalación y, por tanto, una emisora radiofónica. Al dar clic en alguno de los objetos y desplazarlo por todo el gráfico, el escucha interviene alguna de las estaciones. Una estación de radio puede ser intervenida por un solo escucha a la vez y un escucha puede intervenir únicamente una estación de radio a la vez, aunque puede cambiar de estación si lo desea. Las modificaciones que otros radioescuchas realizan durante la intervención también se representan en el gráfico, de esta manera todos los participantes proponen y responden entre ellos. Si

todas las estaciones están intervenidas, el resto de los radioescuchas pueden observar la colaboración de los escuchas activos y esperar a que alguno desocupe alguna de las estaciones.

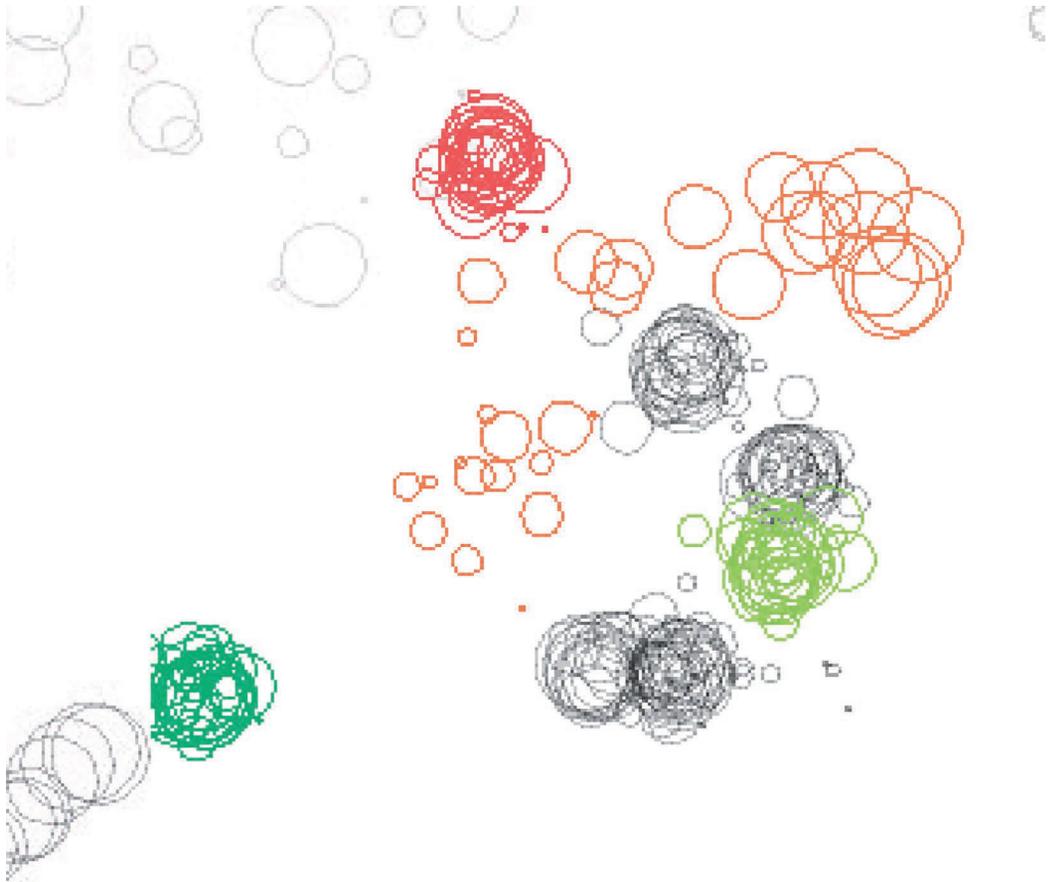


Figura 4.3. | Gráficos colectivos de interactivos de radiobANDA.

El resultado final o mezcla de esta colaboración entre diferentes radioescuchas debe de ser transmitido por la emisora que participa en el proyecto, en este caso Radio Educación. Si la instalación con los radios se encuentra físicamente en la estación de radio, la emisora puede acceder al material directamente; en caso contrario, la emisora puede acceder al material a través de la dirección www.radioBANDA.net/audio.html.

Para fines de valoración del proyecto el público puede acceder a esta última página, sin embargo, la intención es que la emisora transmita esta señal y, por tanto, que los usuarios tengan necesariamente que sintonizar la estación al mismo tiempo que visitan el sitio con el gráfico interactivo.

L a

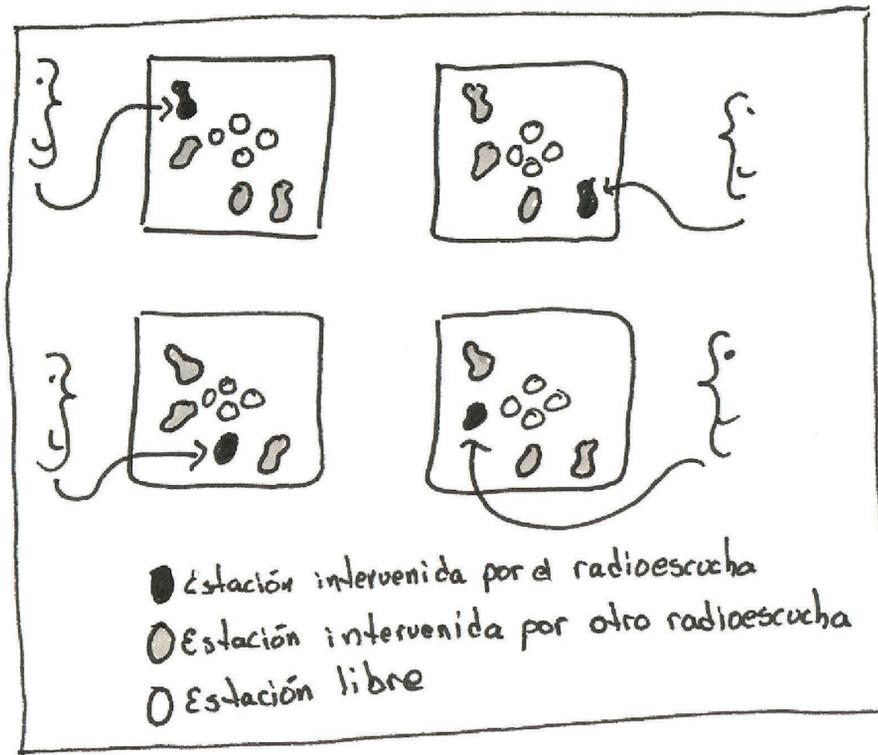


Figura 4.4. E schema de interacción de radiobANDA.

instalación consta de ocho radios de diferentes características, algunos son portátiles, otros son antiguos y hay también un reproductor para automóvil. La intención es mostrar que la radio es accesible y está presente en diferentes momentos y situaciones: en la calle, en el automóvil, en el trabajo, etc. La señal de estos radios es recibida en una computadora por medio de una tarjeta digitalizadora. Cada radio tiene su propio canal de audio. Un “patch” desarrollado en Max/Msp, específicamente diseñado para la instalación, se encarga de realizar las

transformaciones de los diferentes canales de audio en tiempo real. Se decidió emplear síntesis granular, ya que este proceso ofrece una gran gama de transformaciones al tiempo que se preserva la sonoridad del audio original sin su claro reconocimiento. Las transformaciones del audio están determinadas por los trazos que los radioescuchas hacen en el gráfico animado.

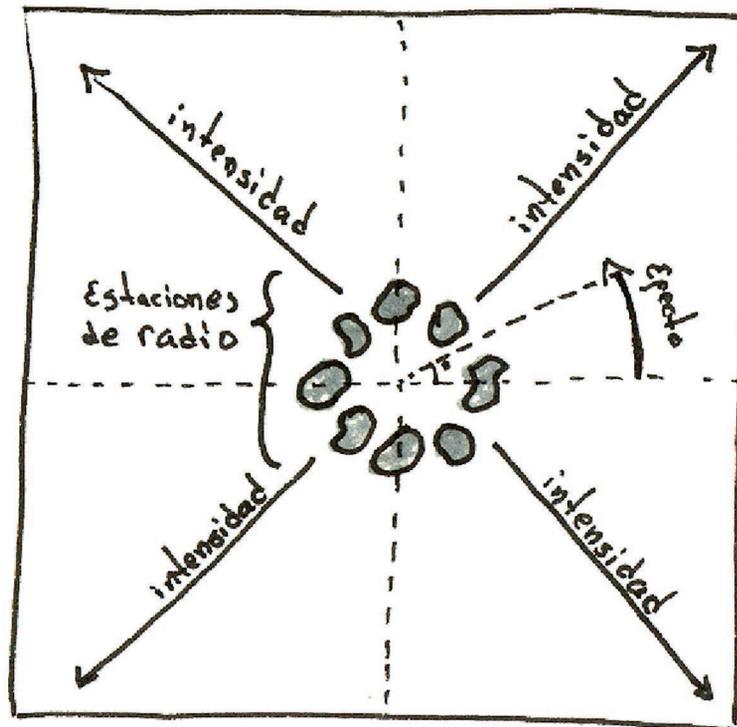


Figura 4.5. | Esquema de control de usuario en radiobANDA.

El gráfico animado es un “applet” desarrollado en Processing y Java y corre sin problema en la mayoría de las computadoras personales, independientemente de su sistema operativo. Generalmente no hay necesidad de instalar o configurar la computadora, pues el gráfico funciona dentro de una página web. En el gráfico se representan varios objetos musicales de diferentes colores. Cada uno de ellos representa una emisora radiofónica. El escucha, como ya se mencionó al principio, modifica algunos parámetros de acuerdo con el desplazamien-

to de los objetos. La intensidad depende de la distancia entre el objeto y el centro del gráfico; el resto de los parámetros dependen del ángulo formado por el lugar original del gráfico y una línea horizontal imaginaria en el centro del gráfico. Este mecanismo le ofrece al radioescucha un control intuitivo del radio que está interviniendo.

La mezcla final de las modificaciones que los escuchas realizan en el gráfico debe ser recibida y transmitida por la estación de radio. Para fines de prueba una copia de esta mezcla es enviada a un servidor Shoutcast, lo que permite escuchar el audio por internet. Es importante recalcar que este proyecto involucra necesariamente a la radio analógica y, por tanto, si bien es cierto que la audiencia podría recibir el audio vía internet, la intención es utilizar la radio tradicional.

Real-time Distributed Media Applications In LANs with OSC

Tristan Jehan, Dan Overholt, Hugo Solís García, Cati Vaucelle
{tristan, dano, hugosg, cati}@media.mit.edu

Abstract

An increasing number of interactive applications must deal with real-time processing, and/or management of large databases of media (i.e., text, sound, images, videos). Often, limitations with computing power, requirements for distributing knowledge, availability and convenience of programming environments, make most intensive systems often fragmented across machines and even locations. These programs must exchange streaming data quickly, and seamlessly.

The Media Laboratory explores and prototypes such systems. However, emphasis is often given on interactivity and content management rather than on communication technology. Open Sound Control (OSC) enables fast prototyping development of efficient communicating programs regardless of the programming language, computer architecture, and operating system.



Toy Symphony
Max/MSP — Max/MSP
www.toy-symphony.net

Large scale electronic performance involving timbral analysis of a symphony orchestra, and sound synthesis controlled by musical toys.

Analysis/Synthesis
Max/MSP — Max/MSP
www.media.mit.edu/~tristan

Perceptually driven timbre models for the Hyperviolin. Sound analysis and additive synthesis are distributed on separate machines.

Passing Glances
Max/MSP — Flash
www.mlc.ie/~cati

Specifically annotated images are retrieved from a large database by SMS messages, and are displayed in real time within a public space.

Graphonic
Max/MSP — Supercollider
www.overtonesystems.com/graphonic

Drawing gesture analysis, and conversion into electronic sounds played back through the interface for aural-tactile feedback. (done at CREATE)

Matrix
C++ — Max/MSP - CSL
Supercollider
www.overtonesystems.com/matrix

Human-computer interface that allows simultaneous control of 144 continuous parameters of an audio synthesis or effects algorithm.

Media Jukebox
Obj-C — Max/MSP
www.media.mit.edu/~tristan

Open media database distributed through locations and time. Simulation of applications for a city network. Presented at Cannes, France.

RadioBANDA
Java — Max/MSP
www.radioBANDA.net

Interactive installation between a Java applet and a collection of physical radios. Users can collaborate on creating FM broadcasted music.

Figura 4.6. | Cartel del proyecto para el primer congreso de Open Sound Control de la Universidad de Berkeley.

AGRADECIMIENTO²

Quiero agradecer a todas las personas que ayudaron durante estos meses en la realización del proyecto, en particular a Tristan Jehan quien ayudó activamente a lo largo del mismo. Gracias al Laboratorio de Medios del MIT por otorgar todas las facilidades para trabajar en este proyecto. Un agradecimiento muy especial a Olaf Matthes (Akustische-kunst, Alemania) por sus asesorías y por desarrollar los objetos para radio en Max/Msp. Agradecemos como siempre a Mary Farbood (Media Lab, USA) por sus asesorías en la compilación de programas, a Brian Whitman (Media Lab, USA) por sus sugerencias y asesorías. Un agradecimiento a la comunidad de creadores de software libre quienes ven en su trabajo una posibilidad artística de beneficio colectivo. Gracias a todas las personas que han desarrollado las herramientas y programas que han hecho posible este tipo de trabajos.³

CÓDIGO

A continuación se presenta a manera de ejemplo el código fuente del Cliente y del Servidor de Java. ■■■

² TOWARD A DEFINITION OF RADIO ART

1. Radio art is the use of radio as a medium for art.
 2. Radio happens in the place it is heard and not in the production studio.
 3. Sound quality is secondary to conceptual originality.
 4. Radio is almost always heard combined with other sounds - domestic, traffic, tv, phone calls, playing children, etc.
 5. Radio art is not sound art - nor is it music. Radio art is radio.
 6. Sound art and music are not radio art just because they are broadcast on the radio.
 7. Radio space is all the places where radio is heard.
 8. Radio art is composed of sound objects experienced in radio space.
 9. The radio of every listener determines the sound quality of a radio work.
 10. Each listener hears their own final version of a work for radio combined with the ambient sound of their own space.
 11. The radio artist knows that there is no way to control the experience of a radio work.
 12. Radio art is not a combination of radio and art. Radio art is radio by artists.
- Manifiesto, <http://www.Kunstradio.at>

³ Este proyecto fue propuesto para la Bienal de Radio en 2002; la pieza no fue seleccionada, pero sirvió para pensar la radio desde una perspectiva interactiva. En 2002 implementar esta tecnología era un trabajo complicado, con la evolución de la radio por internet y el acceso masivo de internet ahora resulta muy simple de implementar. En 2015 se diseñó una nueva versión utilizando un servidor de Meteor que permitiría realizar el ejercicio de manera sencilla. Tal vez en un futuro cercano se pueda activar este trabajo a través de Radio UAM. Por el momento el URL ya no está vigente.

```
1. //
2. // radioBanda_Server.java
3. // radioBanda.Server
4. //
5. // Created by Hugo Solis and Tristan Jehan on Mon Mar 22 2004.
6. // Copyright (c) 2004 MIT Media Lab. All rights reserved.
7. //
8.
9.
10. // -----
11. // IMPORTS
12. // -----
13.
14. import com.ilposed.osc.OSCPort;
15. import com.ilposed.osc.OSCMessage;
16.
17. import java.awt.*;
18. import javax.swing.*;
19. import java.net.ServerSocket;
20. import java.net.InetAddress;
21. import java.util.Vector;
22. import java.awt.event.KeyListener;
23. import java.awt.event.KeyEvent;
24. import java.io.IOException;
25. import java.awt.geom.Point2D;
26.
27.
28. // -----
29. // RADIO BANDA SERVER CLASS
30. // -----
31.
32. public class RadioBanda_Server extends JFrame
33. implements KeyListener {
34.
35.     ServerSocket myServerSocket;
36.     Vector myBuddies;
37.     BuddyListener myBuddyListener;
38.     BuddyAnalyzer myBuddyAnalyzer;
39.     OSCPort myOSCPortA;
40.     OSCPort myOSCPortB;
41.     EmailMsgs emailService;
42.
43.     // -----
44.     // CONSTANTS
45.     // -----
46.
47.     int FREE = -1;
48.     int APPLETT_PORT;
49.     int OSC_PORT;
50.     int NUMBER_CIRCLES = 8;
51.     int CIRCLE_TAKEN = 1000;
```

```

52. int CIRCLE_FREE = 999;
53. static int MAX_NUM_BUDDIES = 20;
54. static String FIRST_COMPUTER;
55. static String SECOND_COMPUTER;
56. int CANVAS_WIDTH = 300;
57. int CANVAS_HEIGHT = 300;
58. int CIRCLE_SIZE = 30;
59. int[] X_LOC = new int[NUMBER_CIRCLES];
60. int[] Y_LOC = new int[NUMBER_CIRCLES];
61. int DIS_MAX = (int) (Math.sqrt(((CANVAS_WIDTH / 2) *
62. (CANVAS_WIDTH / 2)) + ((CANVAS_HEIGHT / 2) *
63. (CANVAS_HEIGHT / 2))));
64.
65. static int IDcounter = 0;
66. int[] x_loc = new int[NUMBER_CIRCLES];
67. int[] y_loc = new int[NUMBER_CIRCLES];
68. int[] taken = new int[NUMBER_CIRCLES];
69.
70.
71. public static void main(String[] args) {
72.     RadioBanda_Server myServer = new RadioBanda_Server();
73.     myServer.show();
74. }
75.
76. public RadioBanda_Server() {
77.     JFrame.setDefaultLookAndFeelDecorated(true);
78.     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
79.     addKeyListener(this);
80.     pack();
81.     try {
82.         readFile();
83.     } catch (IOException e) {
84.         e.printStackTrace();
85.     }
86.
87.     for (int i = 0; i < NUMBER_CIRCLES; i++) {
88.         taken[i] = FREE;
89.     }
90.
91.     // Define the default location of each circle
92.     for (int i = 0; i < NUMBER_CIRCLES; i++){
93.         x_loc[i] = X_LOC[i] = CANVAS_WIDTH / 2 +
94.         (int)(CIRCLE_SIZE * Math.sin(2.0*Math.PI/NUMBER_CIRCLES*i));
95.         y_loc[i] = Y_LOC[i] = CANVAS_HEIGHT / 2 +
96.         (int)(CIRCLE_SIZE * Math.cos(2.0*Math.PI/NUMBER_CIRCLES*i));
97.     }
98.
99.     try {
100.         myServerSocket = new ServerSocket(APPLET_PORT);
101.     } catch (Exception e) {
102.         System.out.println(e.toString());

```

```
103. }
104.
105. try {
106.     // Connect to Macintosh running Max/MSP
107.     myOSCPortA = new
108.     OSCPort(InetAddress.getByName(FIRST_COMPUTER), OSC_PORT);
109.     myOSCPortB = new
110.     OSCPort(InetAddress.getByName(SECOND_COMPUTER), OSC_PORT);
111. } catch (Exception e) {
112.     System.out.println(e.toString());
113. }
114.
115. myBuddies = new Vector();
116. myBuddyListener = new BuddyListener(this);
117. myBuddyAnalyzer = new BuddyAnalyzer(this);
118. myBuddyListener.start();
119. myBuddyAnalyzer.start();
120.
121.     //// init the mail service
122.     emailService = new EmailMsgs();
123. }
124.
125. // Update server with new data
126. public void updateServer(Buddy aBuddy, Object myObject) {
127.     if (((Point)myObject).x % CIRCLE_TAKEN == CIRCLE_FREE) {
128.         x_loc[aBuddy.circle] = X_LOC[aBuddy.circle];
129.         y_loc[aBuddy.circle] = Y_LOC[aBuddy.circle];
130.         taken[aBuddy.circle] = FREE;
131.         updateBuddies(aBuddy.ID, myObject);
132.         updateMAX(aBuddy.circle); // inform MAX
133.     } else {
134.         if (taken[aBuddy.circle] == aBuddy.ID) {
135.             x_loc[aBuddy.circle] = ((Point) myObject).x
136.             % CIRCLE_TAKEN;
137.             y_loc[aBuddy.circle] = ((Point) myObject).y;
138.             updateBuddies(aBuddy.ID, myObject);
139.             updateMAX(aBuddy.circle); // inform MAX
140.         } else if (taken[aBuddy.circle] == FREE) {
141.             x_loc[aBuddy.circle] = ((Point) myObject).x
142.             % CIRCLE_TAKEN;
143.             y_loc[aBuddy.circle] = ((Point) myObject).y;
144.             taken[aBuddy.circle] = aBuddy.ID;
145.             updateBuddies(aBuddy.ID, myObject);
146.             updateMAX(aBuddy.circle); // inform MAX
147.         } else {
148.             aBuddy.circle = FREE;
149.         }
150.     }
151. }
152.
153. // Update all buddies with new data
```

```

154. public void updateBuddies(int anID, Object myObject) {
155.
156.     for (int i=0; i < myBuddies.size(); i++) {
157.         if (((Buddy) (myBuddies.get(i))).ID != anID) {
158.             ((Buddy) (myBuddies.get(i))).updateBuddy(myObject);
159.         }
160.     }
161. }
162.
163. public void readFile() throws IOException {
164.     FIRST_COMPUTER =
165.     ReadFileText.readLine("RadioBanda.dat",0);
166.     SECOND_COMPUTER =
167.     ReadFileText.readLine("RadioBanda.dat",1);
168.     APPLET_PORT = new Integer(ReadFileText.readLine(
169.     "RadioBanda.dat",2)).intValue();
170.     OSC_PORT = new Integer(ReadFileText.readLine(
171.     "RadioBanda.dat",3)).intValue();
172. }
173.
174. // Update Max/MSP with new data
175. public void updateMAX(int aCircle) { // todo updateMax
176.     double distance = Point2D.distance(X_LOC[aCircle],
177.     Y_LOC[aCircle], x_loc[aCircle], y_loc[aCircle]) / DIS_MAX;
178.     if (distance > 1.0) distance = 1.0;
179.     double angle = Math.atan2(y_loc[aCircle] -
180.     Y_LOC[aCircle], x_loc[aCircle] - X_LOC[aCircle]);
181.     // get polar cordiante
182.     if (angle < 0) angle = angle + (2.0 * Math.PI);
183.     // transfer into 2Pi range
184.     angle = (angle + (2.0 * Math.PI / NUMBER_CIRCLES *
185.     aCircle)) + (1.5 * Math.PI);
186.     // rotate according to the circle and match the 0
187.     angle = angle % (2.0 * Math.PI);
188.     // get the final value in 2Pi range
189.     if (angle <= Math.PI) angle = angle / Math.PI;
190.     // convert to 1.0
191.     else angle = Math.abs(((angle - Math.PI) / Math.PI) - 1.0);
192.     // convert to 1.0
193.     angle = 1.0 - angle;
194.     try {
195.         Object[] args = {new Float(angle),
196.         new Float(distance)};
197.         OSCMessage msg = new OSCMessage("/Circle" +
198.         aCircle, args);
199.         if (aCircle < NUMBER_CIRCLES / 2) {
200.             myOSCPortA.send(msg);
201.         } else {
202.             myOSCPortB.send(msg);
203.         }
204.     } catch (Exception e) {

```

```
205.     System.out.println(e.toString());
206.     }
207. }
208.
209. // Quit server
210. public void quitAll() {
211.     for (int i=0; i < myBuddies.size(); i++) {
212.         if (((Buddy) myBuddies.get(i)).mySocket.isConnected()) {
213.             ((Buddy) myBuddies.get(i)).disconnect();
214.         }
215.     }
216.     try {
217.         if (!myServerSocket.isBound()) {
218.             myServerSocket.close();
219.         }
220.     } catch (Exception e) {
221.         System.out.println(e.toString());
222.     }
223.     System.exit(0);
224. }
225.
226. // Send position 0 to everyone
227. public void reset() {
228.     for (int i=0; i < NUMBER_CIRCLES; i++) {
229.         try {
230.             Object[] args = {new Float(0.0), new Float(0.0)};
231.             OSCMessage msg = new OSCMessage("/Circle" + i, args);
232.             if (i < NUMBER_CIRCLES / 2) {
233.                 myOSCPortA.send(msg);
234.             } else {
235.                 myOSCPortB.send(msg);
236.             }
237.         } catch (Exception e) {
238.             System.out.println(e.toString());
239.         }
240.     }
241.     System.out.println("reset");
242. }
243.
244. // Reset Max/MSP audio software
245. public void resetMAX() {
246.     try {
247.         Object[] argsA = {new String("bang")};
248.         OSCMessage msgA = new OSCMessage("/Reset", argsA);
249.         myOSCPortA.send(msgA);
250.         myOSCPortB.send(msgA);
251.     } catch (Exception e) {
252.         System.out.println(e.toString());
253.     }
254.     System.out.println("resetMAX");
255. }
```

```

256.
257. // Manage user input to reset the software
258. //(F1 resets the server, F2 resets Max/MSP)
259. public void keyPressed(KeyEvent e) {
260.     if (e.getKeyCode() == KeyEvent.VK_F1) {
261.         reset();
262.     } else if (e.getKeyCode() == KeyEvent.VK_F2) {
263.         resetMAX();
264.     } else if (e.getKeyCode() == KeyEvent.VK_F3) {
265.         System.out.println("Number of buddies runing: “
266.             +this.myBuddies.size());
267.     }
268. }
269.
270. public void keyReleased(KeyEvent e) {
271. }
272.
273. public void keyTyped(KeyEvent e) {
274. }
275.     }
276.     import java.applet.*;
277.     import java.awt.*;
278.     import java.awt.image.*;
279.     import java.awt.event.*;
280.     import java.io.*;
281.     import java.net.*;
282.     import java.text.*;
283.     import java.util.*;
284.     import java.util.zip.*;
285.     import netscape.javascript.*;
286.     import javax.comm.*;
287.     import javax.sound.midi.*;
288.     import javax.sound.midi.spi.*;
289.     import javax.sound.sampled.*;
290.     import javax.sound.sampled.spi.*;
291.     import javax.xml.parsers.*;
292.     import javax.xml.transform.*;
293.     import javax.xml.transform.dom.*;
294.     import javax.xml.transform.sax.*;
295.     import javax.xml.transform.stream.*;
296.     import org.xml.sax.*;
297.     import org.xml.sax.ext.*;
298.     import org.xml.sax.helpers.*;
299.
300.     public class RadioBanda_Client extends BApplet {
301.         // -----
302.         // RADIO BANDA CLIENT CLASS
303.         // -----
304.
305.
306. // -----

```

```
307. // CONSTANTS
308. // -----
309.
310. int APPLET_PORT = 8081;
311. int NUMBER_CIRCLES = 8;
312. int CANVAS_WIDTH = 300;
313. int CANVAS_HEIGHT = 300;
314. int CIRCLE_SIZE = 30;
315. int CIRCLE_TAKEN = 1000;
316. int CIRCLE_FREE = 999;
317. int SUBCIRCLE_ACTIVE = 9;
318. int SUBCIRCLE_PASSIVE = 4;
319. int NUM_SUBCIRCLES = 30;
320. int GOING_HOME_INC = 5;
321. int FRAME_RATE = 100;
322. int STROKE_WEIGHT = 1;
323. int COLOR_GREY = 250;
324. int[] circleColor = new int[NUMBER_CIRCLES];
325.
326.
327. // -----
328. // GLOBAL VARIABLES
329. // -----
330.
331. Socket mySocket;
332. ObjectOutputStream myOutput;
333. ObjectInputStream myInput;
334. Lector myLector;
335. Vector myVector;
336. BFont myFont;
337. int baseColor;
338. int currentColor;
339. boolean connected;
340. boolean playing;
341. int myCircleIndex;
342. BImage b;
343.
344. int[] x_loc = new int[NUMBER_CIRCLES];
345. int[] y_loc = new int[NUMBER_CIRCLES];
346.
347. //-----
348. //
349. // Create array of circles
350. Circle[] circle = new Circle[NUMBER_CIRCLES];
351.
352. // -----
353. // SETUP FUNCTION
354. // -----
355.
356. void setup() {
357. // General applet setup and initializations
```

```
358. size(300, 300);
359. b = loadImage("background.jpg");
360. myFont = loadFont("Verdana.vlw");
361. textFont(myFont, 48);
362. baseColor = color(COLOR_GREY, COLOR_GREY, COLOR_GREY);
363. currentColor = baseColor;
364. ellipseMode(CENTER_DIAMETER);
365. strokeWeight(STROKE_WEIGHT);
366. framerate(FRAME_RATE);
367. noFill();
368. noSmooth();
369. playing = false;
370.
371. int COLOR_CIRCLE_DEFAULT = color(50, 50, 50, 60);
372.
373. // -----
374. // CIRCLE COLORS
375. // -----
376.
377. // There must be NUMBER_CIRCLES colors
378. int COLOR_CIRCLE_0 = color(249, 128, 67, 220);
379. int COLOR_CIRCLE_1 = color(243, 78, 78, 220);
380. int COLOR_CIRCLE_2 = color(246, 198, 82, 220);
381. int COLOR_CIRCLE_3 = color(225, 233, 94, 220);
382. int COLOR_CIRCLE_4 = color(183, 233, 86, 220);
383. int COLOR_CIRCLE_5 = color(93, 231, 66, 220);
384. int COLOR_CIRCLE_6 = color(64, 237, 129, 220);
385. int COLOR_CIRCLE_7 = color(84, 227, 185, 220);
386.
387. // There must be NUMBER_CIRCLES colors
388. circleColor[0] = COLOR_CIRCLE_0;
389. circleColor[1] = COLOR_CIRCLE_1;
390. circleColor[2] = COLOR_CIRCLE_2;
391. circleColor[3] = COLOR_CIRCLE_3;
392. circleColor[4] = COLOR_CIRCLE_4;
393. circleColor[5] = COLOR_CIRCLE_5;
394. circleColor[6] = COLOR_CIRCLE_6;
395. circleColor[7] = COLOR_CIRCLE_7;
396.
397. // Define the default location of each circle
398. for(int i=0; i<NUMBER_CIRCLES; i++){
399.   x_loc[i] = CANVAS_WIDTH/2 + (int)(CIRCLE_SIZE *
400.     sin(2.0f*PI/NUMBER_CIRCLES*i));
401.   y_loc[i] = CANVAS_HEIGHT/2 + (int)(CIRCLE_SIZE *
402.     cos(2.0f*PI/NUMBER_CIRCLES*i));
403. }
404.
405. // Connection to server
406. try {
407.   mySocket = new Socket(getDocumentBase().getHost(),
408.     APPLET_PORT);
```

```
409. mySocket.setSoTimeout(10000);
410. myOutput = new
411. ObjectOutputStream(mySocket.getOutputStream());
412. myInput = new ObjectInputStream(mySocket.getInputStream());
413. connected = true;
414. myVector = new Vector();
415. myLector = new Lector();
416.
417. } catch (Exception e) {
418.     println("Error while connecting to server!");
419.     background(b);
420.     connected = false;
421.     text("Server Saturated. Please try again later!", 20, 100);
422.     quit();
423. }
424.
425. // Setup circles
426. for (int i=0; i<NUMBER_CIRCLES; i++) {
427.     boolean statusCircle = false;
428.     int sizeCircle = CIRCLE_SIZE;
429.     int defaultColor = COLOR_CIRCLE_DEFAULT;
430.     int highlightColor = circleColor[i];
431.     circle[i] = new Circle(x_loc[i], y_loc[i], sizeCircle,
432.         defaultColor, highlightColor, i, statusCircle);
433. }
434. }
435.
436. // -----
437. // MAIN LOOP
438. // -----
439.
440.
441. void loop() {
442.
443.     if(connected) {
444.         background(b);
445.         for (int i=0; i<NUMBER_CIRCLES; i++) {
446.             circle[i].update();
447.             circle[i].display();
448.         }
449.     }
450. }
451. // -----
452. // Quit function
453. // -----
454.
455. void quit() {
456.     println("Quiting applet!");
457.     try {
458.         if (myOutput != null) myOutput.close();
459.         if (myInput != null) myInput.close();
```

```
460. if (mySocket != null) mySocket.close();
461. connected = false;
462. } catch(IOException e) {
463.     println("Error while quitting applet!");
464. }
465. }
466.
467. public void stop() {
468.     quit();
469. }
470.
471. // -----
472. // CIRCLE CLASS
473. // -----
474.
475.
476. class Circle {
477.
478.     int x, y;
479.     int who;
480.     int size;
481.     int defaultColor;
482.     int highlightColor;
483.     int noiseStrength;
484.     int numSubCircles = NUM_SUBCIRCLES;
485.     float mx[] = new float[numSubCircles];
486.     float my[] = new float[numSubCircles];
487.
488.     boolean taken;
489.     boolean selected = false;
490.     boolean home = true;
491.
492.     // -----
493.     // CONSTRUCTOR
494.     // -----
495.
496.     Circle(int _x, int _y, int _size, int _default,
497.           int _highlight, int _who, boolean _taken) {
498.         x = _x;
499.         y = _y;
500.         size = _size;
501.         who = _who;
502.         defaultColor = _default;
503.         highlightColor = _highlight;
504.         taken = _taken;
505.     }
506.
507.     // -----
508.     // UPDATE
509.     // -----
510.
```

```

511. void update() {
512.   if (taken) {
513.     noiseStrength = SUBCIRCLE_ACTIVE;
514.   } else if (selected()) {
515.     // Update circle location
516.     x = constrain(mouseX, 0, CANVAS_WIDTH);
517.     y = constrain(mouseY, 0, CANVAS_HEIGHT);
518.     noiseStrength = SUBCIRCLE_ACTIVE;
519.     sendCircleLocation(CIRCLE_TAKEN);
520.   } else if (!home) {
521.     noiseStrength = SUBCIRCLE_ACTIVE;
522.     if (x < x_loc[who]) x += GOING_HOME_INC;
523.     if (x > x_loc[who]) x -= GOING_HOME_INC;
524.     if (y < y_loc[who]) y += GOING_HOME_INC;
525.     if (y > y_loc[who]) y -= GOING_HOME_INC;
526.
527.     sendCircleLocation(CIRCLE_TAKEN);
528.     if (x > x_loc[who] - GOING_HOME_INC &&
529.         x < x_loc[who] + GOING_HOME_INC &&
530.         y > y_loc[who] - GOING_HOME_INC &&
531.         y < y_loc[who] + GOING_HOME_INC ) {
532.       x = x_loc[who];
533.       y = y_loc[who];
534.       home = true;
535.       sendCircleLocation(CIRCLE_FREE);
536.       noiseStrength = SUBCIRCLE_PASSIVE;
537.     }
538.
539.   } else if (overCircle(x, y, size)) {
540.     noiseStrength = SUBCIRCLE_ACTIVE;
541.   } else {
542.     noiseStrength = SUBCIRCLE_PASSIVE;
543.   }
544.
545.   // Draw subCircles using Proce55ing
546.   for(int i=1; i<numSubCircles; i++) {
547.     mx[i-1] = mx[i];
548.     my[i-1] = my[i];
549.     mx[i-1] = mx[i] + (int)(random(-noiseStrength,
550.     noiseStrength));
551.     my[i-1] = my[i] + (int)(random(-noiseStrength,
552.     noiseStrength));
553.   }
554.   mx[numSubCircles-1] = x;
555.   my[numSubCircles-1] = y;
556.   mx[numSubCircles-1] = mx[numSubCircles-1] +
557.   (int)(random(-noiseStrength, noiseStrength));
558.   my[numSubCircles-1] = my[numSubCircles-1] +
559.   (int)(random(-noiseStrength, noiseStrength));
560.
561. }

```

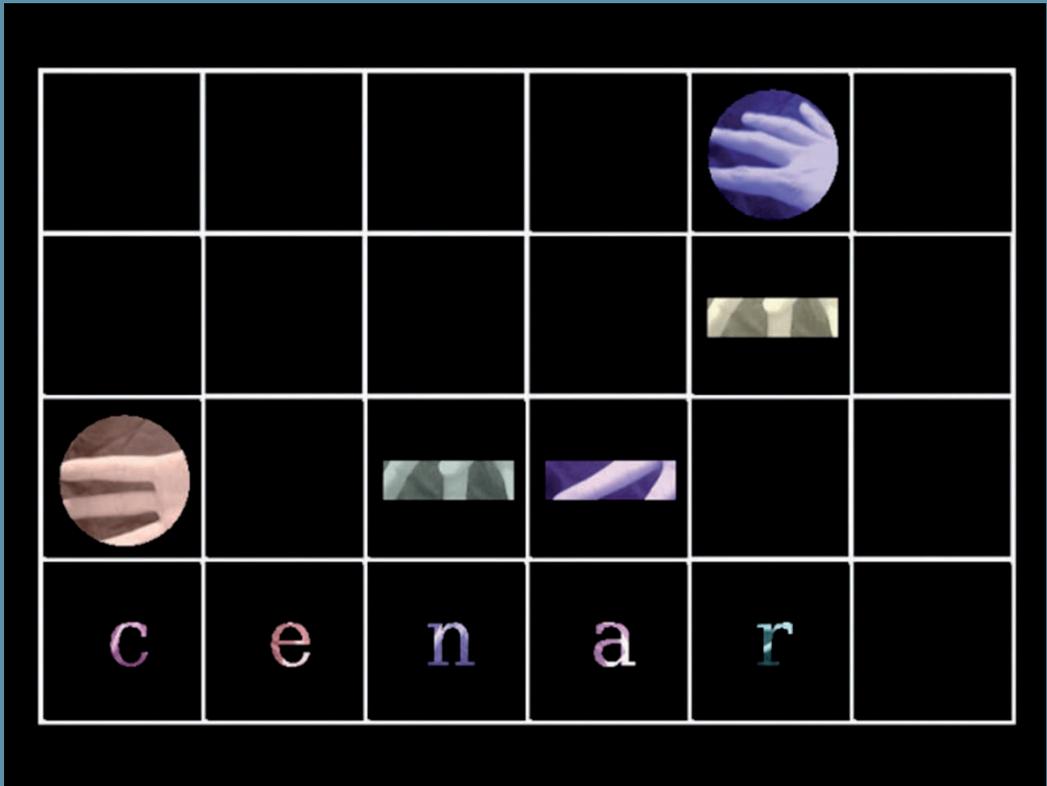
```
562.
563. // -----
564. // SPECIAL METHODS
565. // -----
566.
567. boolean selected() {
568.     if(!taken && overCircle(x, y, size) && mousePressed) {
569.         selected = true;
570.         home = false;
571.         return true;
572.     } else if (selected && mousePressed) {
573.         return true;
574.     } else {
575.         selected = false;
576.         return false;
577.     }
578. }
579.
580. boolean overCircle(int x, int y, int diameter){
581.     float disX = x - mouseX;
582.     float disY = y - mouseY;
583.     if(sqrt(sq(disX) + sq(disY)) < diameter/2) {
584.         return true;
585.     } else {
586.         return false;
587.     }
588. }
589.
590. void sendCircleLocation(int val) {
591.     // Send new circle location
592.     try {
593.         if (val == CIRCLE_FREE) {
594.             myOutput.writeObject(new Point(val +
595.                 (CIRCLE_TAKEN * who), y));
596.         } else {
597.             myOutput.writeObject(new Point(x + (val * who), y));
598.         }
599.         myOutput.flush();
600.         myOutput.reset();
601.     } catch (IOException e) {
602.         println("Problem updating circle location!");
603.     }
604. }
605.
606.
607. // -----
608. // DISPLAY
609. // -----
610.
611.
612. void display() {
```

```

613. if (taken || !home || overCircle(x, y, size)) {
614.   stroke(highlightColor);
615. } else {
616.   stroke(defaultColor);
617. }
618.
619. // draw main circle
620. //ellipse(x, y, size, size);
621.
622. // draw subCircles
623. for(int i=0; i<numSubCircles; i++) {
624.   ellipse(mx[i], my[i], i+1, i+1);
625. }
626. }
627.   }
628.
629.
630. // -----
631. // LECTOR CLASS
632. // -----
633.
634.
635.   class Lector extends Thread {
636.
637.   Lector() {
638.     try {
639.       mySocket.setSoTimeout(0);
640.     } catch(Exception e) {
641.       println("Error with Lector setup!");
642.     }
643.     start();
644.   }
645.
646.   public void run() {
647.     while(connected) {
648.       try {
649.         Object myObject = myInput.readObject();
650.         int bT = ((Point) myObject).x / CIRCLE_TAKEN;
651.         int xT = ((Point) myObject).x % CIRCLE_TAKEN;
652.         int yT = ((Point) myObject).y;
653.
654.         if (xT == CIRCLE_FREE) {
655.           circle[bT].taken = false;
656.         } else {
657.           circle[bT].taken = true;
658.           circle[bT].x = xT;
659.           circle[bT].y = yT;
660.         }
661.       } catch (Exception e) {
662.         connected = false;
663.         println("Error receiving data!");

```

```
664.   quit();
665.   }
666. }
667. }
668.     }
669.
670.     ///  
671.     }
```

Ix-.-.hel

Hoja de sala

Fecha de producción: 2005. *Lugares de exhibición:* Festival Transitio_MX 01; Festival Art and Performance Promenada, Universidad de Washington, Seattle EUA. *Material:* Computadora, teléfonos celulares, proyección de video. *Tecnologías creativas:* Java y JSyn.

DESCRIPCIÓN TÉCNICA

Ix-.-.hel es una instalación audiovisual que genera material visual y sonoro a partir de la traducción e interpretación a código morse de los mensajes de texto que envían los espectadores desde sus teléfonos celulares, así como de imágenes que una cámara de video toma de las manos de los asistentes.

Ix-.-.hel busca crear un lazo visual y auditivo entre tres métodos de comunicación: las manos, que por su naturaleza son vehículo natural de expresión; la clave morse, que fue el primer código de comunicación eléctrico a distancia en tiempo real; y los mensajes instantáneos de los teléfonos celulares que se han adoptado en la cotidianidad de la comunicación contemporánea.

El despliegue de las imágenes en la pantalla con fotografías de la mano en tiempo real (trabajo gestual) propone una correlación directa de la mano fotografiada con el lenguaje de señas. La clave morse traducida a sonido y una imagen en la pantalla contenida en círcu-

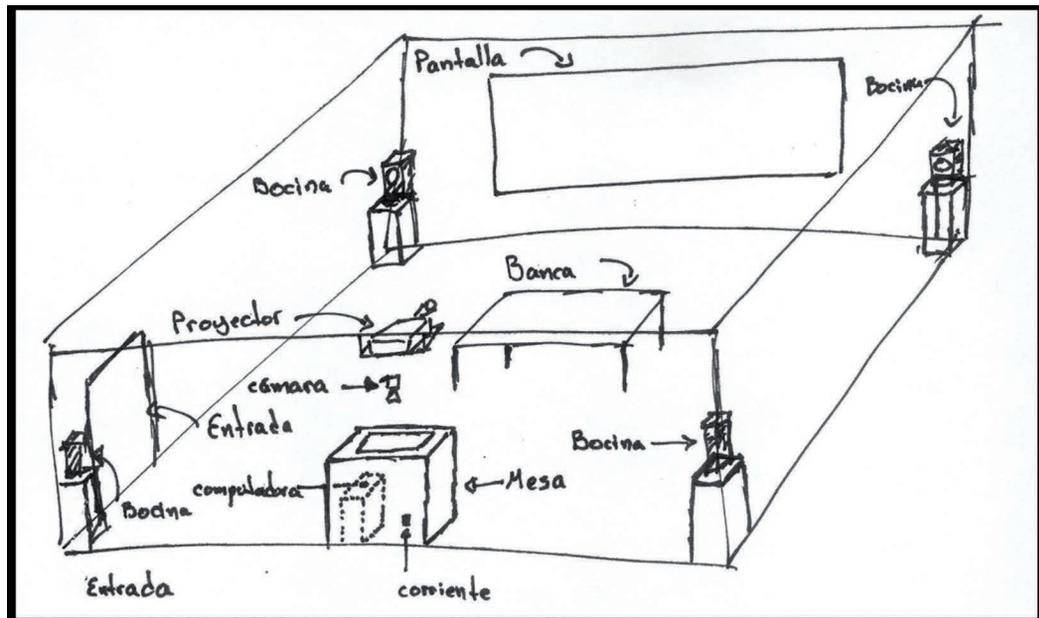


Figura 5.1. | Boceto de la instalación *lx-.-.hel*.

lo-rama, así como la interacción del texto escrito a través del generador de un teléfono móvil, evidencian la red de comunicación de la actualidad y los paralelismos y discordancias de los diferentes métodos de comunicación.

La instalación debe ser montada en un cuarto parcialmente oscurecido. La componente visual es proyectada en una pantalla mediante un proyector y los eventos sonoros son reproducidos por cuatro bocinas colocadas en las esquinas del cuarto.

La instalación funciona con una computadora personal a la que se le conectan una cámara de video por USB o Firewire, un teléfono celular por USB y una tarjeta de audio de cuatro canales. Todo el trabajo de captura de imagen y mensajes de texto, así como la creación del resultado audiovisual final, son realizados por la computadora en la que se está ejecutando un programa en Java específicamente diseñado para la instalación. La salida de imagen es por medio de un proyector y la salida de audio por medio de la tarjeta de audio. La señal de audio debe ser amplificada a una intensidad mediana y reproducida por las bocinas.

Cada vez que un espectador manda un mensaje de texto, éste es convertido a código morse al mismo tiempo que es presentado de forma textual en la parte baja de la proyección. La traducción es utilizada para generar pulsos sonoros e imágenes en forma de puntos o rayas.

Todos los mensajes que se van recibiendo durante la presentación de la instalación son guardados en la computadora y utilizados para generar constantemente eventos audiovisuales siguiendo los principios descritos anteriormente.

La instalación está dedicada a mi hermana Ixchel Solís quien es sorda profunda desde el nacimiento. Los gestos con las manos sean o no del lenguaje de señas han sido importantes vehículos de comunicación entre nosotros desde nuestra infancia. Por su parte, los mensajes de texto de los teléfonos celulares vinieron a ser no sólo un método práctico de comunicación entre nosotros, sino además la vía más utilizada para transmitir emociones instantáneas a distancia.

LA TRANSFIGURACIÓN DIGITAL DE LA TELEGRAFÍA Y SU IMAGINARIO

La tecnología ha sitiado los lenguajes, les ha marcado pautas de producción y ha alterado su sentido de identidad dictando los cánones de producción y belleza. *Ix-.hel* a través del lenguaje computacional Java y sus complementarios en acción (la clave morse, el lenguaje de señas, el lenguaje sonoro y el lenguaje visual) intenta destacar la influencia y las modificaciones que sufre la poética del lenguaje en la actualidad debido al uso inconsciente del material que lo produce. Si consideramos que los teléfonos móviles nos permiten ubicarnos en un espacio-temporal, *Ix-.hel* es un proyecto transdisciplinario que nace bajo la necesidad de tender un vértice entre cuatro lenguajes, sin importar el sitio de la emisión del código inicial (signo) y dando como resultado varios ejes de significados y significantes del lenguaje.

Desde los tiempos en que el motor de madera de un reloj proporcionó la potencia de mover una cinta de papel bajo una pluma y comunicar, el hombre ha estado ante la constante variación de medios y recursos de comunicación. *Ix-.hel* intenta conciliar cuatro distintos lenguajes de comunicación a través del recurso digital con la finalidad de recupe-

rar el sentido lúdico de dos lenguajes relegados (la clave morse y el lenguaje de señas) fusionándolos con el lenguaje sonoro y el lenguaje escrito (texto).

El despliegue de las imágenes en la pantalla con fotografías de la mano en tiempo real (trabajo gestual) propone una correlación directa de la mano fotografiada con el lenguaje de señas. La clave morse traducida a sonido y una imagen en la pantalla contenida en círculo-rama, así como la interacción del texto escrito a través del generador de un teléfono móvil, evidencian la red de comunicación de la actualidad y proponen revalorar el sentido de comunicación en la crisis que vivimos, con el fin de reinterpretar el sentido de comunicación que desde la antigua Grecia se ha tejido con la finalidad de comunicar.

Ix-.hel sugiere la participación de la historia en combinación con la técnica, la consideración del papel de un creador consciente al incluir un contenido claro y directo; y la inclusión de un espectador humanizado a través de la comunicación dirigida de una obra de arte digital.

La experiencia de encontrarse frente a una mano propia que produce gestos en la pantalla y la acción de escribir un texto generador en el teléfono propio sitúan al espectador en la condición de conductor de elementos comunicantes que actúan sobre cuatro ejes de lenguaje. Esta acción directa provoca un sentido de responsabilidad de un espectador cocreador de la obra y lo ubica en un espacio finito de actividades que tienen consecuencias directas con los eventos que percibe en tiempo real.

El resultado electrónico provocado por el mensaje de un celular generador inicial despliega en la pantalla y en el entorno un compendio del imaginario de una tecnología que ha resultado poética: un evento sonoro y visual propositivo que trasfigura la historia y la utiliza con un sentido de contenido abierto que re-considera el uso inconsciente de los recursos digitales.

CÓDIGOS

Este proyecto está conformado por un gran número de clases de Java. A continuación se presenta el código de la clase principal y de la clase `MorseReader` que es la que realiza la conversión de texto a sonido con el apoyo de varias clases secundarias. ■ ■ ■

```
1. package ixchel;
2.
3. import javax.swing.JFrame;
4.
5. import ixchel.utilities.KeyAdapterIxchel;
6. import ixchel.video.Video;
7. import ixchel.audio.JSYNAudio;
8. import ixchel.morse.Morse;
9. import ixchel.sms.*;
10.
11. public class Ixchel extends JFrame {
12.
13.     public DataIxchel data;
14.     public Video video;
15.     public SMSReceiver smsReader;
16.     public WindowIxchel window;
17.     public JSYNAudio jSYNAudio;
18.     public VideoManager videoManager;
19.     public SMSManager smsManager;
20.     public Morse myMorse;
21.
22.     private KeyAdapterIxchel keyAdapter;
23.     private boolean fullScreen = false;
24.
25.     public Ixchel(){
26.         this(false);
27.     }
28.
29.     public Ixchel(boolean fs){
30.
31.         fullScreen = fs;
32.
33.         video = new Video();
34.         video.initVideo();
35.
36.         jSYNAudio = new JSYNAudio();
37.
38.         window = new WindowIxchel(this);
39.         setVisible(true); //required to make the window listen
40.         setSize(100,100);
41.         window.setUp(fullScreen);
42.         System.out.println("initiating window...");
43.         try {
44.             Thread.sleep(5000);
45.         } catch (InterruptedException e) {
46.             e.printStackTrace();
47.         }
48.         myMorse = new Morse();
49.         data = new DataIxchel(this);
50.         data.fillGrid();
51.         smsManager = new SMSManager(this);
```

```

52. smsManager.getText();
53.
54. videoManager = new VideoManager(this);
55. videoManager.getImages();
56.
57. keyAdapter = new KeyAdapterIxchel(this);
58. addKeyListener(keyAdapter);
59. window.addKeyListener(keyAdapter);
60.
61. smsReader = new SMSReceiver(DataIxchel.PORT,
62.   DataIxchel.B_RATE);
63. smsReader.addListener(data);
64.
65. video.startPlayer();
66. videoManager.startVideoGrabber();
67. data.startScanning();
68. window.repaint();
69. }
70.
71. public static void main(String[] args) {
72.   boolean fs = false;
73.   //get javax.command line arguments
74.   for (int x = 0; x < args.length; x++) {
75.     // -fs for full screen
76.     if (args[x].toLowerCase().compareTo("-fs") == 0)
77.       fs = true;
78.   }
79.   new Ixchel(fs);
80. }
81. }

```

```

1. package ixchel;
2.
3. import ixchel.audio.JSYNAudio;
4.
5. import java.awt.Graphics2D;
6. import java.awt.Polygon;
7. import java.awt.Rectangle;
8. import java.awt.geom.Ellipse2D;
9. import java.awt.geom.Rectangle2D;
10. import java.awt.image.BufferedImage;
11. import java.awt.image.LookupOp;
12. import java.awt.image.ShortLookupTable;
13. import java.util.Random;
14.
15. import com.softsynth.jmsl.JMSL;
16. import com.softsynth.jmsl.MusicJob;
17. import com.softsynth.jsyn.*;
18. import com.softsynth.jsyn.circuits.*;
19.
20. public class MorseReader extends MusicJob {

```

```
21.
22. private Ixchel ixchel;
23. private JSYNAudio audio;
24.
25. private String myMorse = "... --- ...";
26. //private String mySMS = "not defined";
27.
28. private int length = 0;
29. private int counter = 0;
30. private static double duration = 60;
31. private double freq_dit = 440;
32. private double dur_dit = 0.2;
33. private double[] enveData_dit;
34. private double freq_dah = 880;
35. private double dur_dah = 0.4;
36. private double[] enveData_dah;
37. private double dur_endWord = 1.0;
38. private double dur_endChar = 2.0;
39. private Random myRandom = null;
40.
41. private SynthEnvelope miSynEnvelope;
42. private EnvelopePlayer miEnvePlayer;
43. private FMPair miFM;
44. private BusWriter miWriterL;
45. private BusWriter miWriterR;
46. private MultiplyUnit miMultyL;
47. private MultiplyUnit miMultyR;
48.
49. public Graphics2D g2;
50. private LookupOp IO;
51. private BufferedImage image = null;
52. private Integer[] coordinates;
53. private int[] myRealLocation;
54. private static final Ellipse2D sDah =
55. new Ellipse2D.Float(10, 10, 100, 100);
56. private static final Rectangle2D sDit =
57. new Rectangle.Float(10, 45, 100, 30);
58. private static final Rectangle2D sWord =
59. new Rectangle.Float(0, 0, 119, 119);
60. private static final int[] xpoints = new int[]{10, 40, 110, 40};
61. private static final int[] ypoints = new int[]{110, 110, 10, 10};
62. private static final Polygon sChar =
63. new Polygon(xpoints, ypoints, 4);
64.
65. private boolean isPlaying = true;
66.
67. public MorseReader(Ixchel ixchel, BufferedImage image,
68. Integer[] coordinates, String mySMS, String myMorse) {
69. super();
70. this.ixchel = ixchel;
71. this.audio = ixchel.jsYNAudio;
```

```

72. g2 = (Graphics2D)ixchel.window.getContentPane().getGraphics();
73. //this.mySMS = mySMS;
74. this.myMorse = myMorse;
75. this.coordinates = coordinates;
76. getLocation(coordinates);
77. myRandom = new Random();
78.
79. length = myMorse.length();
80. setRepeats(999999);
81. setStartTime(JMSL.now());
82.
83. frec_dit = myRandom.nextInt(1600) + 80;
84. frec_dah = frec_dit * (myRandom.nextDouble() + 2);
85.
86. dur_dit = .15 + (myRandom.nextDouble() * .3);
87. dur_dah = dur_dit * 2;
88. dur_endWord = dur_dit * 3;
89. dur_endChar = dur_dit * 4;
90. enveData_dit = new double[] { 0.01, .1, dur_dit - .01, 0.0 };
91. enveData_dah = new double[] { 0.01, .1, dur_dah - .01, 0.0 };
92. generateImage(image);
93. initaudio();
94. }
95.
96. public boolean isPlaying(){
97.     return isPlaying;
98. }
99.
100. private void getLocation(Integer[] twoPoints){
101.     int x = DataIxchel.WIDTH_DEPHASE +
102.     DataIxchel.GRID_LINE_THICK +
103.     (twoPoints[0].intValue() * (DataIxchel.IMAGE_WIDTH +
104.     DataIxchel.GRID_LINE_THICK));
105.     int y = DataIxchel.HIGHT_DEPHASE +
106.     DataIxchel.GRID_LINE_THICK +
107.     (twoPoints[1].intValue() * (DataIxchel.IMAGE_HIGHT +
108.     DataIxchel.GRID_LINE_THICK));
109.     myRealLocation = new int[] {x,y};
110. }
111.
112. private void generateImage(BufferedImage image){
113.     short[][] colorOffset = new short[3][256];
114.     short[] rgb = new short[] {
115.         (short)(myRandom.nextInt(100)-50),
116.         (short)(myRandom.nextInt(100)-50),
117.         (short)(myRandom.nextInt(100)-50)};
118.     for (int i = 0; i < 256; i++) {
119.         colorOffset[0][i] = constrains((short)(i + rgb[0]),
120.         (short)0, (short)255);
121.         colorOffset[1][i] = constrains((short)(i + rgb[1]),
122.         (short)0, (short)255);

```

```

123. colorOffset[2][i] = constrains((short)(i + rgb[2]),
124. (short)0, (short)255);
125. }
126. IO = new LookupOp(new ShortLookupTable(0, colorOffset),
127. null);
128. this.image = new BufferedImage(image.getWidth(),
129. image.getHeight(), BufferedImage.TYPE_INT_RGB);
130. IO.filter(image, this.image);
131. }
132.
133. private short constrains(short value, short min, short max) {
134. if (value < min)
135. value = min;
136. if (value > max)
137. value = max;
138. return value;
139. }
140.
141. private void initaudio(){
142. try {
143. miFM = new FMPair();
144. miWriterL = new BusWriter();
145. miWriterR = new BusWriter();
146. miMultyL = new MultiplyUnit();
147. miMultyR = new MultiplyUnit();
148. miSynEnvelope = new SynthEnvelope(enveData_dit);
149. miEnvePlayer = new EnvelopePlayer();
150.
151. miFM.output.connect(0, miMultyL.inputA, 0);
152. miFM.output.connect(0, miMultyR.inputA, 0);
153. miMultyL.output.connect(0, miWriterL.input, 0);
154. miMultyR.output.connect(0, miWriterR.input, 0);
155. miWriterL.busOutput.connect(0, audio.myReaderL.busInput, 0);
156. miWriterR.busOutput.connect(0, audio.myReaderR.busInput, 0);
157. miEnvePlayer.output.connect(miFM.amplitude);
158.
159. double stereo = myRandom.nextDouble();
160. miMultyL.inputB.set(1.0 - stereo);
161. miMultyR.inputB.set(stereo);
162. miFM.modFrequency.set(myRandom.nextInt(100) + 10);
163. miFM.modAmplitude.set(myRandom.nextInt(10000));
164. } catch (SynthException e) {
165. }
166. }
167.
168. public void start(){
169. startAudio();
170. }
171.
172. private void startAudio() {
173. try {

```

```

174. miFM.start();
175. miMultyL.start();
176. miMultyR.start();
177. miWriterL.start();
178. miWriterR.start();
179. miEnvePlayer.start();
180. } catch (SynthException e) {
181. }
182. }
183.
184. public void stop() {
185.     // try {
186.     //     super.stop(JMSL.now());
187.     // } catch (InterruptedException e) {
188.     //     e.printStackTrace();
189.     // }
190. finish();
191. stopAudio();
192. }
193.
194. private void stopAudio() {
195. try {
196. miFM.stop();
197. miMultyL.stop();
198. miMultyR.stop();
199. miWriterL.stop();
200. miWriterR.stop();
201. miEnvePlayer.stop();
202. } catch (SynthException e) {
203. }
204. }
205.
206. public double repeat(double playTime)
207. throws InterruptedException {
208. setCurrentTime(JMSL.now());
209. if (getCurrentTime() < getStartTime() + duration) {
210. char c = myMorse.charAt(counter % length);
211. switch (c) {
212. case '.':
213. //Audio
214. miFM.frequency.set(freq_dit);
215. miSynEnvelope.write(enveData_dit);
216. miEnvePlayer.envelopePort.queue(miSynEnvelope);
217. //Video
218. g2.translate(myRealLocation[0], myRealLocation[1]);
219. // g2.setClip(sWord);
220. // g2.fill(sWord);
221. g2.setClip(sDit);
222. g2.drawImage(image, null, 0, 0);
223. g2.translate(-myRealLocation[0], -myRealLocation[1]);
224. new drawBlack(dur_dit * .9).launch(JMSL.now());

```

```
225.
226. setRepeatPause(dur_dit);
227. break;
228. case '-':
229. //Audio
230. miFM.frequency.set(frec_dah);
231. miSynEnvelope.write(enveData_dah);
232. miEnvePlayer.envelopePort.queue(miSynEnvelope);
233. //Video
234. g2.translate(myRealLocation[0], myRealLocation[1]);
235. // g2.setClip(sWord);
236. // g2.fill(sWord);
237. g2.setClip(sDah);
238. g2.drawImage(image, null, 0, 0);
239. g2.translate(-myRealLocation[0], -myRealLocation[1]);
240. new drawBlack(dur_dah * .9).launch(JMSL.now());
241.
242. setRepeatPause(dur_dah);
243. break;
244. case ' ':
245. //Video
246. // g2.translate(myRealLocation[0], myRealLocation[1]);
247. // g2.setClip(sWord);
248. // g2.fill(sWord);
249. // g2.translate(-myRealLocation[0], -myRealLocation[1]);
250.
251. setRepeatPause(dur_endWord);
252. break;
253. case '/':
254. //video
255. g2.translate(myRealLocation[0], myRealLocation[1]);
256. // g2.setClip(sWord);
257. // g2.fill(sWord);
258. g2.setClip(sChar);
259. g2.fill(sChar);
260. g2.translate(-myRealLocation[0], -myRealLocation[1]);
261. new drawBlack(dur_endChar * .9).launch(JMSL.now());
262.
263. setRepeatPause(dur_endChar);
264. break;
265. }
266. counter++;
267. return playTime;
268. } else {
269. stopAudio();
270. ixchel.data.locations.add(coordinates);
271. isPlaying = false;
272. finish();
273. return playTime;
274. }
275. }
```

```
276.
277. class drawBlack extends MusicJob {
278. public drawBlack(double time){
279.   setRepeats(0);
280.   setStopDelay(time);
281. }
282.
283. public double stop(double playTime)
284. throws InterruptedException {
285.   g2.translate(myRealLocation[0], myRealLocation[1]);
286.   g2.setClip(sWord);
287.   g2.fill(sWord);
288.   g2.translate(-myRealLocation[0], -myRealLocation[1]);
289.   return playTime;
290. }
291. }
292. }
```




Ojos te vean

Hoja de sala

Fecha de producción: 2007. *Lugar de exhibición:* Festival Transitio_MX 02. *Material:* Servidor web, teléfonos celulares, proyección de video. *Tecnologías creativas:* MaxMsp, Jitter, Java.

DESCRIPCIÓN CONCEPTUAL

O*jos te vean* es un espacio de creación audiovisual colectiva vía internet. Una plataforma que nace como reflexión sobre los procesos y las prácticas de construcción del imaginario colectivo a distancia. Una conjunción simbólica que funciona como eje de un territorio virtual colectivo que aborda la manera en que los imaginarios individuales dialogan, interactúan y se acercan o alejan, a través de una práctica artística conjunta con la intención de producir una redefinición geoidentitaria actual provocada por el uso de la red.

La colaboración telemática en tiempo real como resultado de una colaboración audiovisual entre personas-creadores desde un contexto privado con el fin de trasladarlo al espacio público tiende a igualar la colaboración democratizando la participación creativa. Gloria Sutton señala que una de las aportaciones fundamentales del trabajo colectivo vía network es el privilegio de colaborar sin autoría democratizando la producción y la recepción del arte (Sutton, 2015). Uno de los ejemplos de dichos trabajos se puede encontrar en las producciones de Vitto Accouni, Adrian Piper y Hans Haacke (Alberro y Stimson, 1999).

La posibilidad de desdoblarse en un espectador que se vuelve creador al participar en el proyecto provoca un desprendimiento del biopoder, entendiendo como tal la definición de Foucault. De alguna manera, el usuario-participante se despoja de la disciplina concebida como control social a la hora de participar anónimamente o en privado. El proyecto intenta abolir el control (vigilancia) y la utilidad del material que se genera durante la creación de la obra situando al espectador en el paradigma de verse convertido en actor representante liberado de la acción creativa. *Ojos te vean* concibe el uso de internet no sólo como instrumento de transferencia y distribución de información, sino como “recurso abierto” a la participación artística colectiva.

El proyecto se concibe con dos ejes interconectados: por un lado, la disolución de la frontera geoidentitaria y, por otro, la transposición de la autorrepresentación en un recipiente de confluencia artística del espacio público que incluye el espacio físico y el virtual. La intención de provocar la autorrepresentación y después plasmar el resultado colectivo manipulado por un espectador participante en la obra crea un tejido relacional capaz de combinar e involucrar diferentes estratos de creación conjunta que se metaconducen a la obra final.

Ojos te vean utiliza un compendio de disciplinas que nos adentran en un terreno donde se combinan la imagen, el sonido, la palabra escrita, la abstracción y sobre todo la interactividad colectiva. Aunque el concepto de interactividad es ampliamente utilizado en las ciencias de la comunicación, su campo de aplicación suele comprender el campo performativo del arte. Sheizar Rafaeli ha definido la interactividad como “una expresión extensiva donde una serie de intercambios comunicacionales implican que el último mensaje se relacione con mensajes anteriores a su vez relativos a otros previos” (Sudweeks, McLaughlin y Rafaeli, 1998).

Ojos te vean involucra diferentes niveles de interactividad en los distintos puntos de su creación. Primeramente, el usuario genera aportaciones democráticas de colaboración creativa que pueden ser anónimas o no; segundo, la relación con la máquina-código es capaz de producir la manipulación de los datos de video/audio transformados algorítmicamente; y tercero, la participación colectiva de la audiencia en la representación del cortometraje final donde un espectador puede intervenir en la alteración del subtítulo enviando mensajes de texto por teléfono celular. Al marcar las distintas fases del proyecto podemos inferir que la interactividad es similar al nivel de respuesta en todos los planos de su realización y funciona

como proceso de comunicación en el que cada mensaje se relaciona con el previo y, a su vez, con la relación entre éste y los precedentes.

Ojos te vean es el resultado de provocar un choque dentro de la estética de la representación y una legitimación de un discurso individual que se expande en lo colectivo. Una crítica alterna que ofrece una posibilidad a los modelos de representación masivos con opción de cambiar la realidad previamente plasmada en el video original. Ciertamente, como Gianni Vattimo señala, hemos entrado en la posmodernidad –una especie de “babel informativa”– donde la comunicación y los medios adquieren un carácter central (Vattimo, 2011); un momento que marca la superación de la modernidad dirigida por las concepciones de los modelos cerrados en fundamentos de la historia como huella unitaria del acontecer. La posmodernidad abre el camino a la tolerancia y a la diversidad donde las ideas del pensamiento están estrechamente relacionadas con el desarrollo del escenario multimedia y con la toma de la posición mediática.

El proyecto *Ojos te vean* se origina como acción alternativa a los modelos de comprensión de la identidad, la interactividad, la frontera —el espacio físico y el virtual—, la diversidad y, sobre todo, la territorialidad global. Incluye y propone una estructura relacional que te confronta con una realidad que creías preconcebida. El proyecto intenta generar un modelo que plantea la opción de salir ilesos de una crisis originada por las prácticas cotidianas del uso de la red en la esfera global.

DESCRIPCIÓN TECNOLÓGICA

Ojos te vean es un proyecto que utiliza un gran número de programas y librerías, la mayoría libres y gratuitas para su funcionamiento. La página web fue creada con Drupal y montada sobre un servidor Apache. Desde el sitio www.ojosTeVean.net se descarga un programa específicamente desarrollado para el proyecto que permite a los participantes grabar audio y video con una webcam. La interfase es intuitiva y permite grabar y reproducir el material muy fácilmente. Los materiales grabados por los participantes con este programa son la fuente principal para realizar la parte audiovisual de los trabajos colectivos. Otra fuente del material visual son las imágenes que se pueden buscar con el mismo programa a partir de

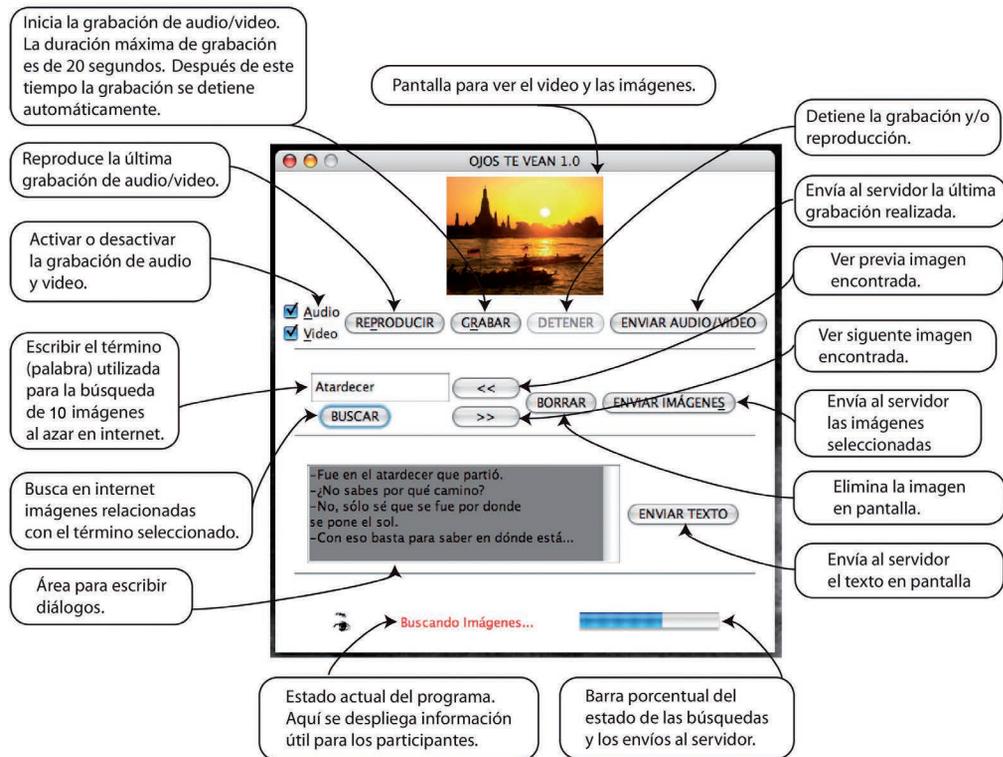


Figura 6.1. | Aplicación web para usuarios de *Ojos te vean*.

palabras o frases. Cada búsqueda produce 10 imágenes aleatorias bajadas de la red. Antes de enviarlas al servidor, el participante puede borrar aquellas que no le interesen. El programa permite también escribir líneas de texto que se utilizan para generar los subtítulos. Con el fin de poder ser utilizado en Windows y Mac, este programa fue elaborado en Java y Java Web Star. Se utilizó la librería JMyron para controlar la webcam, el Yahoo Web Services SDK para la búsqueda de imágenes, SimpleFTP para la transferencia de archivos al servidor y el Java Multimedia Framework para el control, la grabación, la reproducción y la compresión del material.

El servidor del programa fue elaborado en Java/Groovy y se encarga de organizar todo el material que le llega de los participantes además de coordinar los diferentes programas que se utilizan para la generación de los trabajos colectivos. Al recibir nuevo material, el servidor descompacta y clasifica el material en carpetas. Cada noche, si hay más de 20 mate-

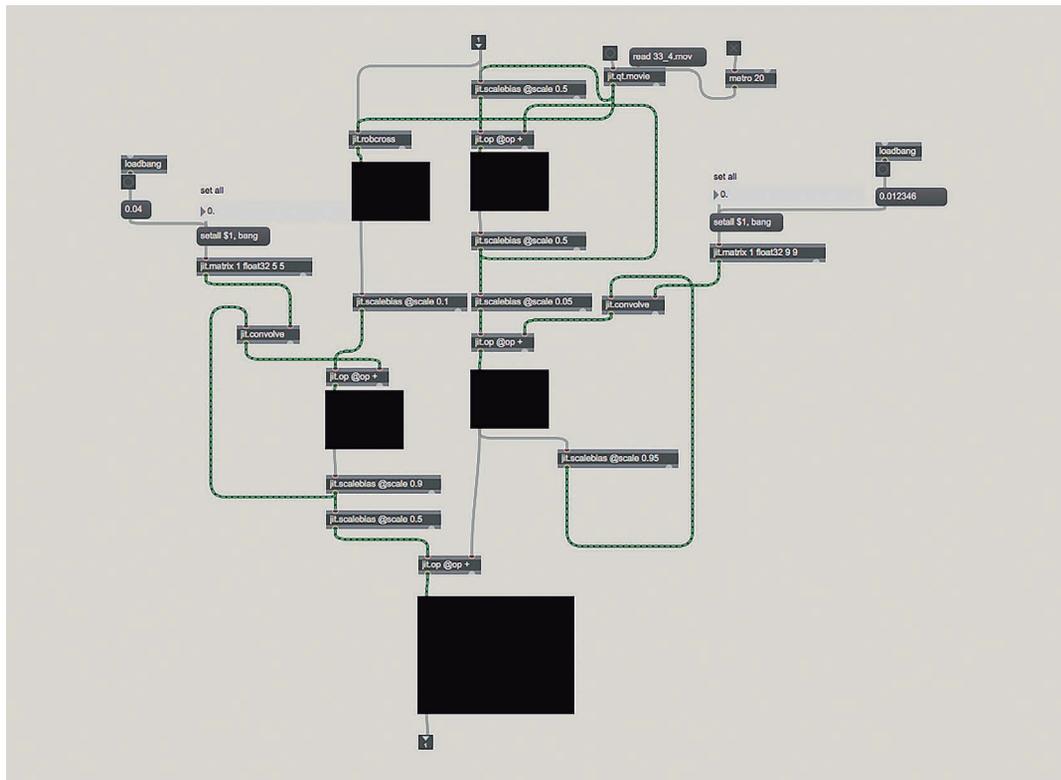


Figura 6.3. | Patch secundario de MaxMSP que realiza el render de video.

subtítulos de los cortos. Otra aplicación en Java/Groovy genera en tiempo los subtítulos a partir del texto previamente recibido y también de nuevo texto que el público envía desde sus teléfonos celulares durante la proyección del corto. Se utilizó la librería SMSLib para controlar el celular que recibe los mensajes y leerlos. Los subtítulos son enviados por el programa vía OSC a Jitter el cual reproduce el video con los subtítulos.

Audio

El audio era agregado al sistema a través de un script de SuperCollider, que se muestra a continuación

```

1. //init
2. (
3.   s = Server.local;
4.   s.boot;
5.   s.notify;
6.   o = Server.local.options;
7.   o.memSize = 4096 * 32;
8.
9.   n = NetAddr(«localhost», 57120); // 57120 is slang default
10.
11. \begin{code}
12. a = OSCResponderNode(nil, "new_score", {
13.   arg time, responder, msg;
14.   [time, responder, msg].postIn;
15.   msg[0].postIn;
16.   msg[1].postIn;
17.   msg[2].postIn;
18.   z = Score.newFromFile(msg[1]);
19.   o = ServerOptions.new.numOutputBusChannels= 2; //monooutput
20.   Score.recordNRT(g, «help-oscFile», msg[2], options: o);
21.   //synthesize
22. }).add;
23. )
24.
25. //synth definition
26. (
27.   SynthDef(«springSounds», {arg dur=10, warp=0.1, freq = 1.0,
28.     buffer = 0, amp = 0.5, output = 0;
29.     var src, env, local;
30.     env = EnvGen.kr(Env.new([0.001, 1.0, 1.0, 0.001], [dur * 0.05,
31.       dur * 0.9, dur * 0.05], «exponential»), dur, doneAction: 2);
32.     src = Warp0.ar(buffer, warp, freq, 0.1, 8, 0.01, 4);
33.     local = LocalIn.ar(2) * 0.99;
34.     local = DelayN.ar(local, 0.3, 0.3);
35.     local = (local * 0.5) + src;
36.     LocalOut.ar(local);
37.     Out.ar(output, local * env * amp);
38.   }).load(s);
39.
40.   SynthDef("extendedSounds", {arg dur=10, var1=0.1, freq = 1.0,
41.     buffer = 0, amp = 0.5, output = 0;
42.     var src, env, out, local;
43.     env = EnvGen.kr(Env.new([0.001, 1.0, 1.0, 0.001], [dur * 0.1,
44.       dur * 0.7, dur * 0.2], «exponential»), dur, doneAction: 2);
45.     src = Warp0.ar(buffer, var1, freq, 0.1, 8, 0.01, 4);
46.     local = LocalIn.ar(2) * 0.99;
47.     local = DelayN.ar(local, 0.3, 0.3);
48.     local = (local * 0.5) + src;
49.     LocalOut.ar(local);
50.     Out.ar(output, local * env * amp);
51.   }).load(s);

```

```

52.
53. SynthDef("grainedSounds", {arg dur=10, freq=1.0, buffer=0,
54. amp = 0.5, output = 0, var1 = 0.5;
55. var src, env, trate, durG, clk, local;
56. env = EnvGen.kr(Env.new([0.001, 1.0, 1.0, 0.001], [dur * 0.1,
57. dur * 0.7, dur * 0.2], «exponential»), dur, doneAction: 2);
58. trate = (var1 * 15) + 4;
59. durG = 2 / trate;
60. clk = Impulse.kr(trate);
61. src = TGrains.ar(2, Impulse.ar(trate), buffer, freq,
62. TRand.kr(0, BufDur.kr(buffer)), durG,
63. WhiteNoise.kr(0.6), 0.1, 4);
64. local = LocalIn.ar(2) * 0.99;
65. local = DelayN.ar(local, 0.3, 0.3);
66. local = (local * 0.5) + src;
67. LocalOut.ar(local);
68. Out.ar(output, local * env * amp);
69. }).load(s);
70. )
71.
72. //stuff
73. //s.sendBundle(0.1, ["b_allocRead", 0,
74. "/Users/hugosg/Documents/ojosTeVean/data/support/1.wav"]);
75. s.sendBundle(0.1, ["b_allocRead", 0,
76. "/Volumes/staging/research/ojosTeVean/data/audio/47_0.wav"]);
77. s.sendBundle(0.1, ["s_new", "springSounds", -1, 0, 0, "dur", 120,
78. "freq", 1.0, "amp", 0.8, "buffer", 0]);
79. s.sendBundle(0.1, ["s_new", "grainedSounds", -1, 0, 0, "dur", 10,
80. "freq", 1.0, "amp", 0.8, "buffer", 0, "var1", 0.9]);
81. s.sendBundle(0.1, ["s_new", "extendedSounds", -1, 0, 0, "dur", 20,
82. "freq", 0.5, "amp", 0.8, "buffer", 0, "var1", 1.5]);
83. a.remove;
84.
85. //test
86. (
87. x = [
88. [2.99999, ["s_new", "longSounds", -1, 0, 0, "dur", 10]],
89. [10.0, ["c_set", 0, 0]]
90. ];
91.
92. o = ServerOptions.new.numOutputBusChannels = 2;
93. Score.recordNRT(x, "help-oscFile.osc",
94. «/Users/hugosg/Documents/ojosTeVean/data/audioMix/test.aiff»,
95. options: o); // synthesize
96. )

```

Render y control del servidor

A continuación se muestra la clase Java del servidor que se encarga de generar los videos y controlar la base de datos. ■ ■ ■

```
1. package ojosTeVeanServer;
2.
3. import java.io.BufferedReader;
4. import java.io.FileInputStream;
5. import java.io.FileNotFoundException;
6. import java.io.FileOutputStream;
7. import java.io.IOException;
8. import java.io.InputStream;
9. import java.io.InputStreamReader;
10. import java.io.OutputStream;
11. import java.io.OutputStreamWriter;
12. import java.net.InetSocketAddress;
13. import java.util.Calendar;
14. import java.util.Date;
15. import java.util.Iterator;
16. import java.util.List;
17. import java.util.Timer;
18. import java.util.TimerTask;
19. import de.sciss.net.OSCClient;
20. import de.sciss.net.OSCMessage;
21.
22. import scoreGenerator.ScoreGenerator;
23. import scoreGenerator.ScoreGenerator2;
24.
25. public class MoviesGenerator {
26.
27.     Timer myTimer;
28.     SQLconnection sQLconnection;
29.
30.     public MoviesGenerator(){
31.         myTimer = new Timer();
32.         sQLconnection = new SQLconnection();
33.         System.out.println("movies generator created");
34.     }
35.
36.     public void startDailyProcess(){
37.         /*start the processes to run every
38.         night starting this nighth at 23:30 */
39.         System.out.println("startDailyProcess");
40.         Calendar today = Calendar.getInstance();
41.         Calendar todayAtNight = Calendar.getInstance();
42.         todayAtNight.set(
43.             today.get(Calendar.YEAR),
44.             today.get(Calendar.MONTH),
```

```

45. today.get(Calendar.DATE),
46. 23, 35, 0);
47. myTimer.scheduleAtFixedRate(new GenerateMovie(),
48. todayAtNight.getTime(), 1000 * 60 * 60 * 24);
49. // myTimer.scheduleAtFixedRate(new GenerateMovie(),
50. //10 * 1000, 1000 * 60 * 60 * 4);
51. }
52.
53. class GenerateMovieTest extends TimerTask{
54. public void run() {
55. System.out.println("testing clock");
56. }
57. }
58.
59. class GenerateMovie extends TimerTask{
60. public void run() {
61. Calendar rightNow = Calendar.getInstance();
62. String fileName = rightNow.get(Calendar.YEAR) + "-" +
63. rightNow.get(Calendar.MONTH) + "-" +
64. rightNow.get(Calendar.DAY_OF_MONTH) + "-" +
65. rightNow.get(Calendar.HOUR_OF_DAY) + "-" +
66. rightNow.get(Calendar.MINUTE);
67. System.out.println("Building piece: " + fileName);
68.
69. String parent =
70. Server.serverProperties.getProperty("DATA_FOLDER");
71. InfoParser ip = new InfoParser(parent,
72. Roman.toRoman(Server.moviesCounter));
73.
74.
75. /* THIS PART GENERATES THE VIDEO, IT HAS SEVERAL STEPS*/
76. /*CREATING TEXT PART AND ENDING*/
77. MovieTextCreator mtcI = new MovieTextCreator(parent +
78. "/videoMix/" + fileName + "-INIT.mov");
79. mtcI.setText((String[])ip.getInitText(), fileName,
80. "INIT", parent);
81. mtcI.createMovie();
82.
83. MovieTextCreator mtcE = new MovieTextCreator(parent +
84. "/videoMix/" + fileName + "-END.mov");
85. mtcE.setText((String[])ip.getEndingText(), fileName,
86. "END", parent);
87. mtcE.createMovie();
88.
89. /*CREATING THE JITTER PART */
90. ip.createVideoTextFiles(fileName);
91. try {
92. OSCClient client = OSCClient.newUsing( OSCClient.UDP);
93. client.setTarget(new InetSocketAddress("127.0.0.1", 7400));
94. client.start();
95. client.send(new OSCMessage("/ojosMovie",

```

```

96.     new Object[] {fileName}));
97.     System.out.println("sending message to Jitter");
98.     } catch (IOException e) {
99.         e.printStackTrace();
100.    }
101.
102.    try {
103.        System.out.println("sleeping to give time for jitter to f");
104.        Thread.sleep(1000 * 60 * 15); // half an hour
105.    } catch (InterruptedException e1) {
106.        e1.printStackTrace();
107.    }
108.
109.    /*INTERMEDIATE FILES INTO MPG FORMAT*/
110.    try {
111.        ProcessBuilder pb = new ProcessBuilder("/ffmpeg/ffmpeg",
112.            "-i",
113.            parent + "videoMix/" + fileName + "-INIT.mov",
114.            "-sameq",
115.            parent + "videoMix/" + fileName + "-INIT.mpg");
116.        pb.redirectErrorStream(true);
117.        Process process = pb.start();
118.        InputStream is = process.getInputStream();
119.        InputStreamReader isr = new InputStreamReader(is);
120.        BufferedReader br = new BufferedReader(isr);
121.        String line;
122.        System.out.println("cre mpg from mov of INIT " + fileName);
123.        while ((line = br.readLine()) != null) {
124.            System.out.println(line);
125.        }
126.        process.destroy();
127.        System.out.println("done with mpg cr of INIT " + fileName);
128.    } catch (IOException e) {
129.        e.printStackTrace();
130.    }
131.
132.    try {
133.        ProcessBuilder pb = new ProcessBuilder("/ffmpeg/ffmpeg",
134.            "-i",
135.            parent + "videoMix/" + fileName + "-END.mov",
136.            "-sameq",
137.            parent + "videoMix/" + fileName + "-END.mpg");
138.        pb.redirectErrorStream(true);
139.        Process process = pb.start();
140.        InputStream is = process.getInputStream();
141.        InputStreamReader isr = new InputStreamReader(is);
142.        BufferedReader br = new BufferedReader(isr);
143.        String line;
144.        System.out.println("creat mpg from mov of END " + fileName);
145.        while ((line = br.readLine()) != null) {
146.            System.out.println(line);

```

```

147. }
148. process.destroy();
149. System.out.println("done with mpg crea of END " + fileName);
150. } catch (IOException e) {
151.     e.printStackTrace();
152. }
153.
154. try {
155.     ProcessBuilder pb = new ProcessBuilder("/ffmpeg/ffmpeg",
156.         "-i",
157.         parent + "videoMix/" + fileName + "-BODY.mov",
158.         "-sameq",
159.         parent + "videoMix/" + fileName + "-BODY.mpg");
160.     pb.redirectErrorStream(true);
161.     Process process = pb.start();
162.     InputStream is = process.getInputStream();
163.     InputStreamReader isr = new InputStreamReader(is);
164.     BufferedReader br = new BufferedReader(isr);
165.     String line;
166.     System.out.println("crea mpg from mov of BODY " + fileName);
167.     while ((line = br.readLine()) != null) {
168.         System.out.println(line);
169.     }
170.     process.destroy();
171.     System.out.println("done with mpg cre of BODY " + fileName);
172. } catch (IOException e) {
173.     e.printStackTrace();
174. }
175.
176. /*GLUE FILES*/
177. FileInputStream fisINIT;
178. FileInputStream fisBODY;
179. FileInputStream fisEND;
180. FileOutputStream fosALL;
181. try {
182.     fisINIT = new FileInputStream(parent +
183.         "videoMix/" + fileName + "-INIT.mpg");
184.     fisBODY = new FileInputStream(parent +
185.         "videoMix/" + fileName + "-BODY.mpg");
186.     fisEND = new FileInputStream(parent +
187.         "videoMix/" + fileName + "-END.mpg");
188.     fosALL = new FileOutputStream(parent +
189.         "videoMix/" + fileName + "-ALL.mpg");
190.     byte[] buf = new byte[1024];
191.     int len;
192.
193.     try {
194.         while ((len = fisINIT.read(buf)) > 0) {
195.             fosALL.write(buf, 0, len);
196.         }
197.     } catch (IOException e) {

```

```

198. e.printStackTrace();
199. }
200.
201. try {
202.     while ((len = fisBODY.read(buf)) > 0) {
203.         fosALL.write(buf, 0, len);
204.     }
205. } catch (IOException e) {
206.     e.printStackTrace();
207. }
208.
209. try {
210.     while ((len = fisEND.read(buf)) > 0) {
211.         fosALL.write(buf, 0, len);
212.     }
213. } catch (IOException e) {
214.     e.printStackTrace();
215. }
216. } catch (FileNotFoundException e) {
217.     e.printStackTrace();
218. }
219.
220. try {
221.     ProcessBuilder pb = new ProcessBuilder("/ffmpeg/ffmpeg",
222.         "-i",
223.         parent + "videoMix/" + fileName + "-ALL.mpg",
224.         parent + "videoMix/" + fileName + "-ALL.mov");
225.     pb.redirectErrorStream(true);
226.     Process process = pb.start();
227.     InputStream is = process.getInputStream();
228.     InputStreamReader isr = new InputStreamReader(is);
229.     BufferedReader br = new BufferedReader(isr);
230.     String line;
231.     System.out.println("creating HD version " + fileName);
232.     while ((line = br.readLine()) != null) {
233.         System.out.println(line);
234.     }
235.     process.destroy();
236.     System.out.println("done with HD version " + fileName);
237. } catch (IOException e) {
238.     e.printStackTrace();
239. }
240.
241. /*Generate audio*/
242. String[] theNames = ip.getSoundsForAudio(); //getnames
243. ScoreGenerator2 sg = new ScoreGenerator2(parent, fileName,
244.     theNames);
245. sg.generateScore();
246.
247. int mem = 32768 * 32;
248. try {

```

```
249. Process process = new ProcessBuilder(
250.     "/Applications/SuperCollider_f/scsynth",
251.     "-N",
252.     parent + "scores/" + fileName + ".osc",
253.     "_",
254.     parent + "audioMix/" + fileName + ".wav",
255.     "44100",
256.     "WAV",
257.     "int16",
258.     "-o",
259.     "2",
260.     "-m",
261.     mem + "").start();
262.
263. InputStream is = process.getInputStream();
264. InputStreamReader isr = new InputStreamReader(is);
265. BufferedReader br = new BufferedReader(isr);
266. String line;
267. System.out.println("start rendering in sc of " + fileName);
268. while ((line = br.readLine()) != null) {
269.     System.out.println(line);
270. }
271. process.destroy();
272. System.out.println("done with rende in sc of " + fileName);
273. } catch (IOException e) {
274.     e.printStackTrace();
275. }
276.
277. /*ADING AUDIO*/
278. try {
279.     ProcessBuilder pb = new ProcessBuilder("/ffmpeg/ffmpeg",
280.     "-i",
281.     parent + "audioMix/" + fileName + ".wav",
282.     "-i",
283.     parent + "videoMix/" + fileName + "-ALL.mov",
284.     parent + "videoHD/" + fileName + "-HD.mov"
285.     );
286.     pb.redirectErrorStream(true);
287.     Process process = pb.start();
288.     InputStream is = process.getInputStream();
289.     InputStreamReader isr = new InputStreamReader(is);
290.     BufferedReader br = new BufferedReader(isr);
291.     String line;
292.     System.out.println("star with audio adding " + fileName);
293.     while ((line = br.readLine()) != null) {
294.         System.out.println(line);
295.     }
296.     process.destroy();
297.     System.out.println("done with audio adding" + fileName);
298. } catch (IOException e) {
299.     e.printStackTrace();
```

```
300. }
301.
302. /*CREATING WEB SIZE VERSION*/
303. try {
304.     ProcessBuilder pb = new ProcessBuilder("/ffmpeg/ffmpeg",
305.         "-i",
306.         parent + "videoHD/" + fileName + "-HD.mov",
307.         "-s", "cif",
308.         parent + "videoWeb/" + fileName + "-WEB.mov");
309.     pb.redirectErrorStream(true);
310.     Process process = pb.start();
311.     InputStream is = process.getInputStream();
312.     InputStreamReader isr = new InputStreamReader(is);
313.     BufferedReader br = new BufferedReader(isr);
314.     String line;
315.     System.out.println("starting web version " + fileName);
316.     while ((line = br.readLine()) != null) {
317.         System.out.println(line);
318.     }
319.     process.destroy();
320.     System.out.println("done with web version" + fileName);
321. } catch (IOException e) {
322.     e.printStackTrace();
323. }
324.
325. /****** PUTING ON THE SERVER *****/
326. //putVideo on server
327. sQLconnection.addingMovie(fileName + "-WEB.mov",
328.     fileName + "-HD.mov", ip.getTitle());
329. //if eveything good new counter for movie
330. Server.moviesCounter++;
331. }
332. }
333.
334. public static void main(String[] args) {
335.     MoviesGenerator mg = new MoviesGenerator();
336.     mg.startDailyProcess();
337. }
338. }
```




Exhalation-water

Hoja de sala

Fecha de producción: 2009. *Lugar de exhibición:* Universidad de Washington. *Material:* Laptops, circuitos electrónicos, *iTouches*. *Tecnologías creativas:* Pure Data, Java, Audio Analyzer.

RESUMEN

Exhalation-water es una experiencia sonora que invita al participante a realizar un recorrido auditivo paralelo al que su propio calor realiza a través de los ductos del aire acondicionado en arquitecturas de múltiples espacios. Muchos complejos arquitectónicos transportan el calor en conductos de agua formando una red. El agua portadora del calor es enfriada en una unidad centralizada. En este recorrido el participante porta un *iTouch* que, utilizando un software diseñado para esta experiencia, amalgama un ambiente sonoro transformado en tiempo real con grabaciones manipuladas del lugar. La dupla sonora conformada por una realidad alterada y la ficción produce una curva dramática que sustenta el reencuentro del calor propio. A lo largo del trayecto el participante se encuentra con varios objetos en los que debe exhalar aire caliente para poder mantener la experiencia activa y recordar poéticamente que es su calor el que transita por todos los espacios.

CONCEPTO

Exhalation-Water es una instalación interactiva que propone desde una plataforma multidimensional el reencuentro con la poética de transitar un espacio apropiándose de su sonoridad hasta llegar al instante primigenio. Surge como pregunta que se desvanece dentro de lo posible. *Exhalation-Water* es un espacio que recorre el halo generado por un aliento que no se contiene. Una exhalación provocada intencionalmente que transcurrirá vía digital-objeto-transformación-disolución de la esencia para convertirse en la materialización sonora que transcurre y que es depositaria de la sustancia original. Regresar al origen, el agua como factor central del discurso, agua sin límites y al mismo tiempo depositario total.

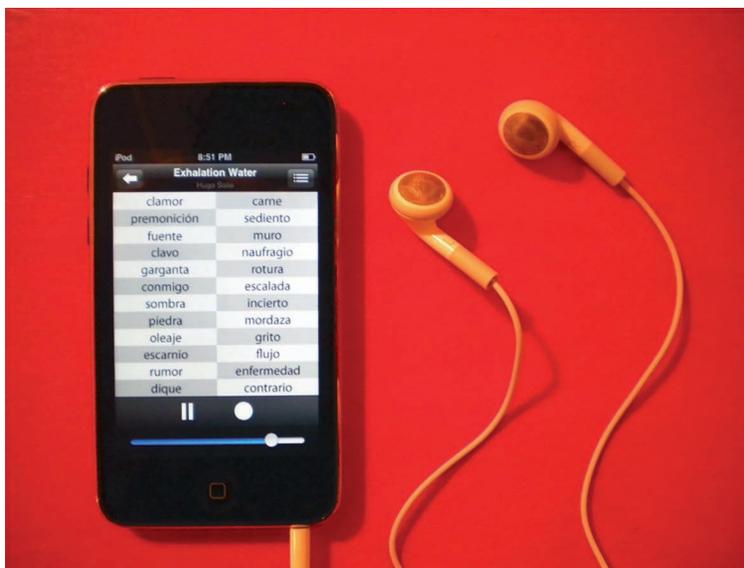


Figura 7.1. | Aplicación para dispositivo móvil para el proyecto *Exhalation-Water*.

¿No nos disolvemos cotidianamente dentro de la ciber-realidad sin ser capaces de apropiarnos y transformar ese impulso inicial en escenarios posibles de una práctica de identidad tecnologizada? ¿No es el vapor la esencia primigenia que nos contiene? La instalación no ofrece respuesta alguna, deja abierta la posibilidad de encontrarse con el espacio desnudo (sin manipulación) y al mismo tiempo ofrece la oportunidad de transitar el terreno

con una banda sonora que se mezcla y mimetiza con los sonidos que acompañan el recorrido. *Exhalation-Water* pretende la apropiación digital del momento poético; el despliegue que sucede al momento de nuestra exhalación cuando nos vemos desdoblados en un discurso sonoro/mecánico y nos transportamos hasta los audífonos de un *iTouch*. El desplazamiento múltiple. Nosotros aliento, nosotros audio y nosotros vehículo de transporte. Caminar la realidad que se transforma. El andar como experiencia estética. Es importante aclarar que el objeto que nos acompaña y guía desde su interfase (un *iTouch*) no es casual y cobra sentido. El *iTouch* representa el objeto identitario de la mundialización de los objetos prácticos: globalización pura y mercantil. La poética transportada por los instantes localizados en un punto que se torna material, pero que al mismo tiempo es indefinido en un punto fijo. Paradoja de localización. El espectador que transita por un recorrido alterno y llega al origen primigenio y desnudo. Nuestro destino el –agua final– ofrece dos posibilidades: o bien escuchar el agua en su devenir natural o preferir el discurso tecnológico y quedarnos con el *iTouch* encendido. Esa decisión del espectador implica una reflexión o una acción automática. Por un lado preferir un ambiente sonoro que se encuentra dado en un escenario natural, o bien, escoger el discurso alterno de una realidad que adquiere un valor autónomo: una sonoridad que al ser legitimada en su existencia digital cobra sentido de realidad.

El proyecto fue la parte artística del examen de calificación para el programa de doctorado en Artes Digitales y Medios Experimentales. Conociendo el interés del autor en desarrollar un viaje sonoro, el comité formuló el siguiente cuestionamiento:

Crear una pieza que pueda adaptarse dinámicamente a diferentes entornos utilizando el sonido como su principal medio de comunicación (no necesariamente sólo a través del audio, otros medios de producción de sonido también podrían ser utilizados). Se le pide que configure dos instancias de trabajo de su pieza en dos sitios en el campus de su elección. Estos dos casos de su pieza deben absorber la mayor cantidad de información posible de su entorno, lo que refleja aspectos destacados de los sitios. Finalmente, los dos sitios deben ser paradas en un paseo que usted tomará, comenzando y terminando en Raitt Pasillo. Este paseo en el campus, al igual que una experiencia cinematográfica, debe ser un viaje poético basado en el tiempo en el que los dos sitios están integrados, teniendo el sonido como su componente central.

Si bien usted debe estar completamente a cargo de la producción de este proyecto, le invitamos a pedir ayuda para hacer frente a parte de su logística, especialmente el traslado de equipos a los sitios y el establecimiento de las cosas (es posible que también quiera tener personas que le ayuden durante el paseo, el equipo de supervisión que queda en los sitios, etc.).



Figura 7.2. | Mapa de tuberías para el diseño del recorrido.

También debe tener un pequeño presupuesto para componentes electrónicos u otras cosas necesarias para completar su proyecto.



Figura 7.3. | Caja de análisis de audio del proyecto *Exhalation-Water* dentro de un elevador en uw.

DESCRIPCIÓN TECNOLÓGICA

Para generar la poética de *Exhalation-Water* esta pieza contiene varios elementos técnicos, software y hardware. El primer elemento de la obra consiste en una compleja escena sonora creada para este proyecto en Pure Data que se ejecuta en el *iTouch/iPhone* a través del programa *rjdj*. Este momento sonoro reproduce y transforma en tiempo real sonidos pregrabados del lugar al mismo tiempo que manipula el entorno sonoro recibido por el micrófono. El material pregrabado sirve para construir la composición sonora y las transformaciones del entorno sirven para producir una realidad alterada que provoca la atención en sonidos que son inadvertidos en la escucha normal. La escena sonora contiene 24 situaciones acústicas diferentes que evolucionan composicionalmente de acuerdo con el estadio del recorrido.

Un segundo elemento se compone por las cajas negras que se colocan en puntos estratégicos del lugar de acuerdo con su arquitectura, significado y uso. Éstas contienen en su interior una computadora que se encarga de extraer información del entorno. Cada cinco minutos, un programa diseñado para este proyecto analiza el entorno sonoro del lugar y extrae su variación temporal en intensidad acústica y su variación temporal en calidad tímbrica. Estos valores son procesados, segmentados y clasificados y nos permiten inferir propiedades y situaciones del lugar tales como actividades realizadas por la audiencia, cantidad de personas, abrir y cerrar de puertas o elevadores y uso de celulares, entre otros.

Los valores obtenidos del análisis acústico se traducen en una de 24 palabras posibles que se despliega en la pantalla de la caja. La palabra producida por este objeto analizador del entorno sonoro sólo se despliega en la pantalla si el participante exhala aire caliente en el orificio elíptico de la caja en la cual hay un sensor de temperatura y un sensor de humedad. Estos sensores se activan únicamente al percibir un cambio abrupto en sus valores. De no recibir cambios, la caja presenta únicamente la leyenda “Estoy escuchando el entorno”. La finalidad del gesto de exhalar aire es estimular en el participante la atención en su calor y la transferencia de su calor al entorno.

Finalmente, la palabra desplegada en la pantalla de la caja se utiliza para seleccionar el entorno sonoro adecuado en el *iTouch/iPhone*. La selección de la escena sonora se realiza al presionar en la pantalla del *iTouch/iPhone* la misma palabra que aparece en la pantalla de la caja. Las palabras seleccionadas pertenecen a un conjunto variado de palabras neutras que subjetivamente propician el juego lúdico y la metáfora.

Utilizando estos recursos técnicos, el participante es invitado a recorrer una trayectoria sonoro-metafórica similar y/o paralela a la que tienen los ductos de aire acondicionado en grandes complejos de edificios y que en muchas ocasiones convergen en una central de enfriamiento. En caso de existir esta central, ésta deberá ser el lugar de arribo de la travesía sonora en donde metafóricamente el viajante se reencontrará con su calor exhalado en los otros puntos del recorrido. ■ ■ ■



Tell-Tale Piano

Hoja de sala

Fecha de producción: 2008. *Lugares de exhibición:* McLeod Residence. *Material:* Piano, madera, motores y circuito electrónico y microcontrolador. *Tecnologías creativas:* Groovy, AvrStudio.

DESCRIPCIÓN

En el cuento *El corazón delator* de Edgar Allan Poe el autor describe cómo el narrador mata al anciano con quien vive y en esta descripción incluye la narración de cómo entierra sus restos desmembrados en el suelo de su cuarto. Después de deshilar el relato, el narrador culpable confiesa su asesinato a la policía por el hecho de no soportar el sonido de los latidos del corazón que salen debajo del piso donde el cuerpo está escondido.

Aunque la obra *Tell-Tale Piano* (piano delator) no es programática, tanto la instalación como la composición sonora están basadas en el cuento de Poe utilizando un antiguo piano transformado electrónicamente y mecánicamente para representar el imaginario sonoro que se describe en el cuento.

El recorrido de transformar un piano abandonado (obtenido sin costo) en una instalación sonora propone reciclar y convertir una historia en la que el espectador es invitado a sentar-



Figura 8.1. | Desensamblaje del piano vertical.

se en una silla con la finalidad de mantener el instrumento musical vivo y latiendo, lo cual representa el corazón que se describe en el cuento.

La pieza fue compuesta sobreponiendo capas. En este proceso cada capa fue colocada con un tempo diferente, como un homenaje al compositor Conlon Nancarrow (quien renovó el significado de las pianolas con sus estudios). La pieza de 17 minutos contiene tres secciones donde cada una tiene una textura distinta. En la primera sección masas de sonidos se sobreponen con tempos cuyas proporciones se desfasan lenta y gradualmente produciendo diferentes patrones rítmicos en cada ciclo. En la segunda sección se crea un momento dramático agregando gradualmente capas con tempos más y más veloces. En la última sección un sutil juego de capas independientes (cada una en constante acelerando y retardando) producen nuevamente cambios de patrón en constante movimiento.

Perceptualmente, el objetivo de la instalación es invitar al espectador a apreciar los sonidos desde el suelo, ayudando no sólo a oír el sonido sino además a sentirlo con los pies y la parte baja del cuerpo.

DESARROLLO DE LA OBRA

Lo primero que busqué fue hacer un trabajo con sonido analógico. Una vez que decidí que se trabajaría con un cuerpo de piano vertical procedí a buscar uno en el mercado de segunda mano. Una vez que se obtuvo se desmanteló en el taller de trabajo removiendo la cama de las teclas y toda la maquinaria obteniendo así un objeto rectangular con las cuerdas libres y la caja de resonancia intacta.

Adquirí diversos tipos de solenoides de distintos tamaños y características y de manera más o menos intuitiva después de probar todos los motores escogí el modelo que mejor se ajustaba a las necesidades. Es importante recalcar que la pieza no había sido inicialmente concebida para permanecer horizontal, sin embargo los solenoides con los que contaba no tenían sistema de recarga o regreso y, por tanto, se requería la fuerza de gravedad para que regresaran a su punto de partida.¹ Posteriormente adquirí una barra de aluminio y tornillos prisioneros Allen del tamaño necesario para sujetar los motores. Realicé las mediciones necesarias para distribuir 24 motores en todo el registro del piano de manera equidistante con la intención de cubrir todo el registro de frecuencias.

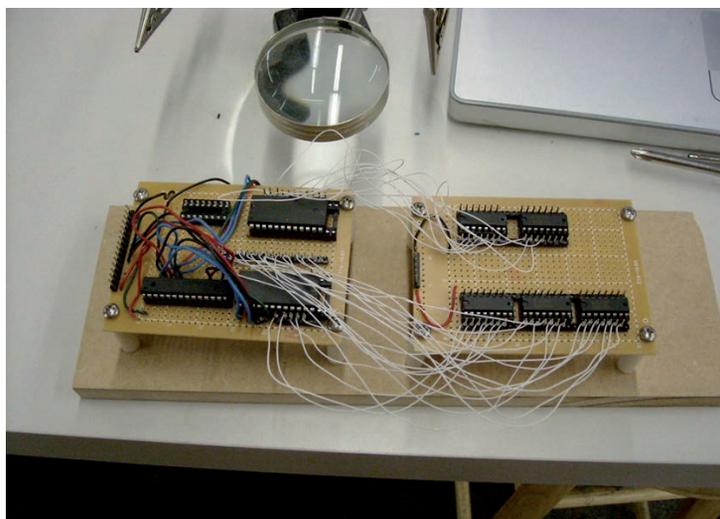


Figura 8.2. | Placa de circuito con *rapid prototyping* para la pieza *Tell-Tale Piano*.

¹ Es interesante notar el diálogo que se dio entre concepto e implementación, ya que una restricción tecnológica detonó una estética.

Esta fue mi primera instalación sonora con componentes electromecánicas. Como tal, fue una exploración y un proceso de aprendizaje interesante que dio pie a una serie de obras en donde el material era generado por medios analógicos pero controlados y sincronizados de manera digital. Fue también una extraordinaria manera de aprender y aplicar el conocimiento, en cuanto a la programación de microcontroladores, y una de las intenciones fue hacer un trabajo encapsulado que no requiriera computadora externa, por tal motivo y sabiendo que la pieza era estática los ciclos sonoros debían estar grabados en la microcomputadora.

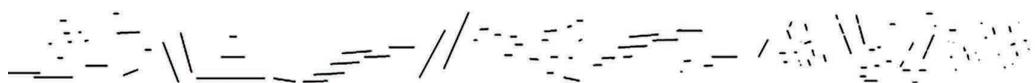


Figura 8.3. | Partitura gráfica de la pieza *Tell-Tale Piano*.

Probablemente la contribución más interesante del proyecto radica en la manera en que se produjo la partitura y la posibilidad de insertarla en la memoria del microcontrolador. Para este proyecto no se buscaba un trabajo de producción algorítmica y por tal motivo la creación de la partitura como producto de una serie de procedimientos lógicos estaba descartada. Opté por diseñar la estructura musical en la computadora y posteriormente traducirla a una secuencia numérica que se pudiera almacenar en el EPROM del microcontrolador. Al haber la intención de producir gestos orgánicos y bajo la influencia de los rollos de pianola de Nancarrow decidí que el primer paso sería dibujar la partitura. Utilicé un editor de imagen convencional para realizar el trazado de las trayectorias sonoras haciendo una correspondencia en donde la presencia de color significaría activación de motores y la parte blanca silencio. Una vez realizada la partitura gráfica realicé un código en el lenguaje de programación Groovy que leía el mapa de bits del archivo de imagen y lo convertía en un arreglo numérico que se almacenaba en un archivo de texto. Posteriormente, dicho arreglo se copiaba al archivo *.h* del proyecto del microcontrolador. Esta técnica de producción de material estático para sistemas electrónicos fijos me permitió generar gestos sonoros orgánicos y considerarla una buena opción para futuros proyectos.

A continuación se presenta el código de *Groovy* que convierte la imagen original de los trazos en el arreglo numérico y, posteriormente, en segmento del código numérico genera-

do. Como se puede observar en la parte numérica la imagen blanca se convierte en 0 y la presencia de color en 1. Existe también, en la última versión, una segunda lista que define la modulación en tiempo de toda la partitura.

```

1. package pianolaScorer
2. import javax.imageio.ImageIO
3.
4. class ScoreGenerator {
5.   def static final MAX_SOLENOIDS = 32
6.
7.   static void main(args) {
8.     println "this program writes scores" +
9.     "for the array of solenoids"
10.    println "arguments are: 1) sourceImage" +
11.    " for pitches, max height 32"
12.    println "2) sourceImage for dynamics, « +
13.    "similar dimensions that image 1"
14.    println "3) name of file to write, " +
15.    ".h for themega168"
16.    println "4) min value of modulation," +
17.    " 5) max value of modulation (0-255)"
18.
19.    def maxModulation = args[4].toInteger()
20.    def minModulation = args[3].toInteger()
21.    def rangeModulation = maxModulation - minModulation
22.    def outFile = new File(args[2])
23.    def myImageDynamics = ImageIO.read(new File(args[1]))
24.    def myImage = ImageIO.read(new File(args[0]))
25.
26.    outFile.write("")
27.    println "image size = " +
28.    myImage.height + " * " + myImage.width
29.    outFile.append("const uint16_t pieceSize = «
30.    + myImage.width + ";\n")
31.
32.    //for pitch information
33.    outFile.append("const uint32_t pieceData[] PROGMEM = {\n")
34.    (0..<myImage.width).each{column ->
35.      def solenoids = ""
36.      (0..<myImage.height).each{row ->
37.        def color = myImage.getRGB(column, row)
38.        def colorAverage = (int)((color >> 16 & 0xFF)
39.        + (color >> 8 & 0xFF) + (color >> 0 & 0xFF)) / 3)
40.        if (colorAverage > 254) solenoids += "0"
41.        else {
42.          solenoids += "1"
43.        }
44.      }
45.      outFile.append("0b" + solenoids + ("0"

```

```

46.  * (MAX_SOLENOIDS - myImage.height)) + ",\n")
47.  }
48.  outFile.append("};\n")
49.
50.  //for dynamics information
51.  outFile.append("const uint8_t» +
52.  "pieceModulation[] PROGMEM = {\n")
53.  (0..<myImageDynamics.width).each{column ->
54.  def intensity = minModulation
55.  (0..<myImageDynamics.height).each{row ->
56.  def color = myImageDynamics.getRGB(column, row)
57.  def colorAverage = (int)(((color >> 16 & 0xFF)
58.  + (color >> 8 & 0xFF) + (color >> 0 & 0xFF)) / 3)
59.  if (colorAverage < 255) intensity =
60.  (int)((myImageDynamics.height - row)
61.  * rangeModulation / myImageDynamics.height
62.  + minModulation)
63.  }
64.  outFile.append(intensity + ",\n")
65.  }
66.  outFile.append("};\n")
67.  }
68.  }
69.  const uint16_t pieceSize = 23;
70.  const uint32_t pieceData[] PROGMEM = {
71.  0b10000000000000000000000000000000,
72.  0b01000000000000000000000000000000,
73.  0b00100000000000000000000000000000,
74.  0b00010000000000000000000000000000,
75.  0b00001000000000000000000000000000,
76.  0b00000100000000000000000000000000,
77.  0b00000010000000000000000000000000,
78.  0b00000001000000000000000000000000,
79.  0b00000000100000000000000000000000,
80.  0b00000000010000000000000000000000,
81.  0b00000000001000000000000000000000,
82.  0b00000000000100000000000000000000,
83.  0b00000000000010000000000000000000,
84.  0b00000000000001000000000000000000,
85.  0b00000000000000100000000000000000,
86.  0b00000000000000010000000000000000,
87.  0b00000000000000001000000000000000,
88.  0b00000000000000000100000000000000,
89.  0b00000000000000000010000000000000,
90.  0b00000000000000000001000000000000,
91.  0b00000000000000000000100000000000,
92.  0b00000000000000000000010000000000,
93.  0b00000000000000000000001000000000,
94.  };
95.  const uint8_t pieceModulation[] PROGMEM = {
96.  100,

```

```

97. 100,
98. 100,
99. 100,
100.    100,
101.    100,
102.    100,
103.    100,
104.    100,
105.    100,
106.    100,
107.    100,
108.    100,
109.    100,
110.    100,
111.    100,
112.    100,
113.    100,
114.    100,
115.    100,
116.    100,
117.    100,
118.    100,
119.    };

```

La secuencia numérica grabada en el EPROM del microcontrolador era leída en *loop* y los valores eran utilizados para seleccionar los momentos de actividad o reposo de cada motor. Contar con un código de estas propiedades permitía también cambiar la velocidad general de la pieza. A continuación se presenta el código de la pieza almacenado en el microcontrolador.

```

1.  /*****
2.  This program plays a piece in 30 pwm solenoids
3.  *****/
4.
5.  #include <avr/io.h>    // provides AVR port and register
6.  #include <avr/interrupt.h> // for the interrupts
7.  #include <avr/pgmspace.h> // from PROGMEM
8.  #include <stdlib.h>    // because of the random
9.  #define F_CPU 1000000UL // has a clock speed of 1 MHz
10. #include <util/delay.h> // uses F_CPU to provide delay
11.
12. #include "pieceData.h" //file with data
13. // #include "pieceDataTest.h"
14.
15. #define serialInput 0 //define serialinput from mega168to74f575a
16. #define chipSelect 1 // define chip select input
17. #define shiftClock 2 // define shift clock pulse input

```

```

18. #define storeClock 3 // define store clock pulse input
19. #define readWrite 4 // define read write input
20.
21. #define SOLENOIDS 23 // number of solenoids
22. /*low numbers in PERIOD makes
23. ppppp higher make ffff but more noise*/
24. #define PERIOD 10 // max 255, set the speed of the pwm
25. #define METRONOME 10 // max 255, set the metro speed
26. #define MULTIPLIER 1
27.
28. uint32_t values = 0;
29. uint8_t ticks = 0;
30. uint16_t currentMoment = 0;
31. uint8_t multiplierCounter = 20;
32.
33. //interrupt ofr pwm the motors
34. ISR(TIMER0_COMPA_vect)
35. {
36. if(ticks == 0) ticks = 1;
37. else ticks = 0;
38. }
39.
40. //interrupt of metronome
41. ISR(TIMER2_COMPA_vect)
42. {
43. multiplierCounter++;
44. if(multiplierCounter > MULTIPLIER){
45. multiplierCounter = 0;
46. currentMoment++;
47. if(currentMoment >= pieceSize) currentMoment = 0;
48. }
49. }
50.
51. void atmega168init(void)
52. {
53. //timer for PWM
54. TIMSK0 |= (1 << OCIE0A); //setting a timer with compare match
55. OCR0A = PERIOD; // time compare
56. TCCR0A |= (1 << WGM01); // Clear on Timer Compare Match
57. //TCCR0B |= (1 << CS02); //prescaler 256
58. TCCR0B |= (1 << CS01) | (1 << CS00); //prescaler 64
59. //TCCR0B |= (1 << CS01); //prescaler 8
60. //timer for piece control
61. TIMSK2 |= (1 << OCIE2A);
62. OCR2A = METRONOME;
63. TCCR2A |= (1 << WGM21);
64. TCCR2B |= (1 << CS22) | (1 << CS21) | (1 << CS20); //prescaler 1024
65.
66. //define the pins as outputs
67. DDRC |= (1 << serialInput) | (1 << chipSelect) |
68. (1 << shiftClock) | (1 << storeClock) | (1 << readWrite);

```

```

69. PORTC&=~((1<<serialInput)|(1<<chipSelect)|
70. (1<<shiftClock)|(1<<storeClock)|(1<<readWrite));
71.
72. sei(); //enable interrupts
73. }
74.
75. //send the serial data
76. void sendByteToRegister(uint32_t data)
77. {
78.   uint32_t i, temp;
79.   temp=data;
80.   PORTC&=~_BV(serialInput); // sets datapin to output a LOW
81.   PORTC&=~_BV(chipSelect); // put chipSelect to output a LOW
82.   PORTC&=~_BV(storeClock); // put storeClock to output a LOW
83.   PORTC|=_BV(readWrite); // put readWrite to output a HIGH
84.
85.   for (i=0;i < 32;i++) //write 32 bit to shift register
86.   {
87.     PORTC|=(data&1);
88.     _delay_us(10);
89.     PORTC|=_BV(shiftClock); // sets clockpin to output a HIGH
90.     _delay_us(10);
91.     PORTC&=~_BV(shiftClock); // sets clockpin to output a LOW
92.     _delay_us(10);
93.     PORTC&=~_BV(serialInput); // sets datapin to output a LOW
94.     _delay_us(10);
95.     data=temp>>1;
96.     temp=data;
97.   }
98. }
99.
100. //set the array as parallel output
101. void storeRegister()
102. {
103. PORTC&=~_BV(chipSelect); // put chipSelect to output a LOW
104. PORTC&=~_BV(readWrite); // put read write to output a LOW
105. _delay_us(10);
106. PORTC&=~_BV(storeClock); // sets storeClock to output a LOW
107. _delay_us(10);
108. PORTC|=_BV(storeClock); // sets storeClock to output a HIGH
109. _delay_us(10);
110. }
111.
112. int main()
113. {
114. atmega168init();
115. while (1) // loop here forever
116. {
117. if(ticks == 1)
118.   values = pgm_read_dword(&pieceData[currentMoment]);
119. else values = 0; // turn motors off

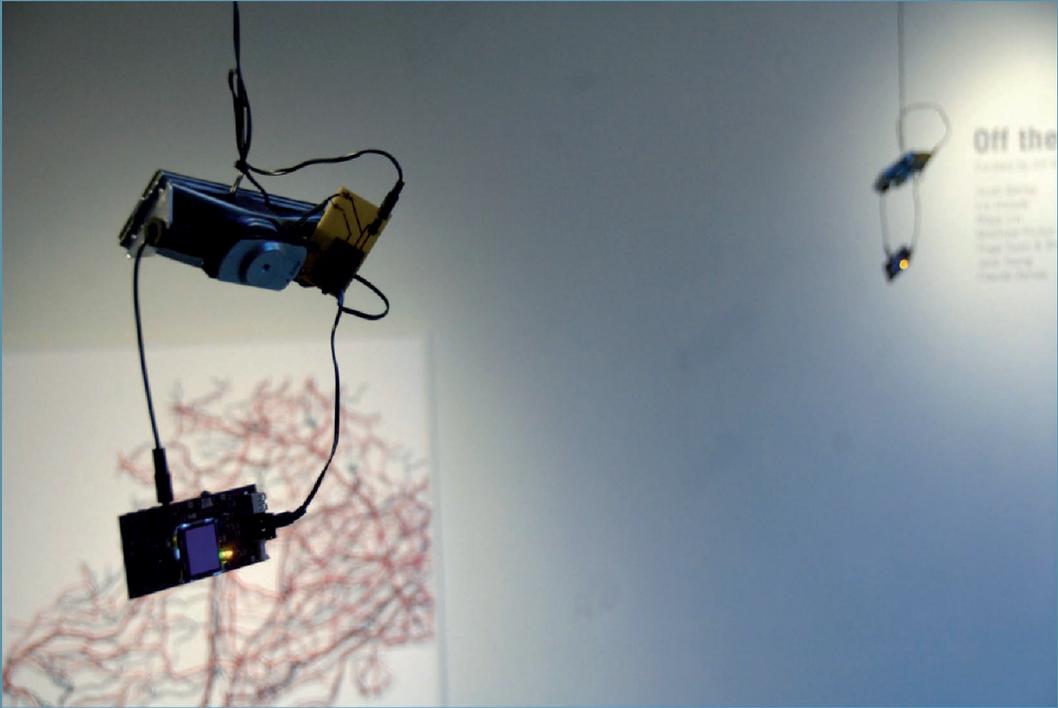
```

```
120. sendByteToRegister(values);
121. storeRegister();
122. OCR0A = pgm_read_byte(&pieceModulation[currentMoment]);
123. }
124. return 0;
125. }
```

En cuanto a la electrónica esta obra fue realizada en una placa fenólica y las conexiones realizadas con la técnica de *rapid prototyping*. Un microcontrolador AVR168 contenía el código necesario para decodificar la partitura, controlar los motores y alojar la partitura. Un par de circuitos de corrimiento (*Shift Register*) fueron utilizados para convertir los datos secuenciales de la salida del AVR en un control independiente para los 24 motores. Las salidas en paralelo de los circuitos de corrimiento controlaban un arreglo de transistores en otra placa fenólica en donde se encontraba el circuito de potencia. Es importante mencionar que toda la pieza funcionaba con una fuente de poder para computadora. Debido a que los motores funcionaban con 12 volts y los circuitos integrados con 5 volts la solución más estable y económica fue utilizar una fuente de poder para ATX de 600 Watts. ■■■



Figura 8.4. | Distribución de motores para la obra *Tell-Tale Piano*.



Off the

October 10, 2011

Paul Smith

Lee Smith

Walter Smith

Michael Smith

David Smith

John Smith

Thomas Smith

Swarms

Hoja de sala

Fecha de producción: 2008. *Lugar de exhibición:* Kirkland Art Center. *Material:* Bocinas, amplificadores, SuperBirds microcomputadoras. *Tecnologías creativas:* Pure Data.

DESCRIPCIÓN

S*warm Intelligence* describe el comportamiento de alto nivel producido por la interacción de varias unidades descentralizadas no jerárquicas no complejas de inteligencia de bajo nivel con la interacción local a corto plazo. Los ejemplos tradicionales para describir sistemas de enjambre son colonias de insectos, bandadas y escuelas de peces cuyo comportamiento general es más sofisticado que el de cada uno de los miembros aislados.

Siguiendo dicho paradigma este proyecto modela un sistema de enjambre que emplea una colección de circuitos electrónicos llamados *superbirds*. Los *superbirds* fueron diseñados y desarrollados en la Universidad de Washington por Bruce Hemingway y su equipo. Desde su creación se han utilizado en varios cursos como medio para enseñar sistemas integrados de ordenador y también como dispositivos para proyectos artísticos. Su conjunto de capacidades como la síntesis de audio, la pantalla a color y la comunicación por radio las convierten en una tecnología adecuada para explorar experiencias audiovisuales interactivas.

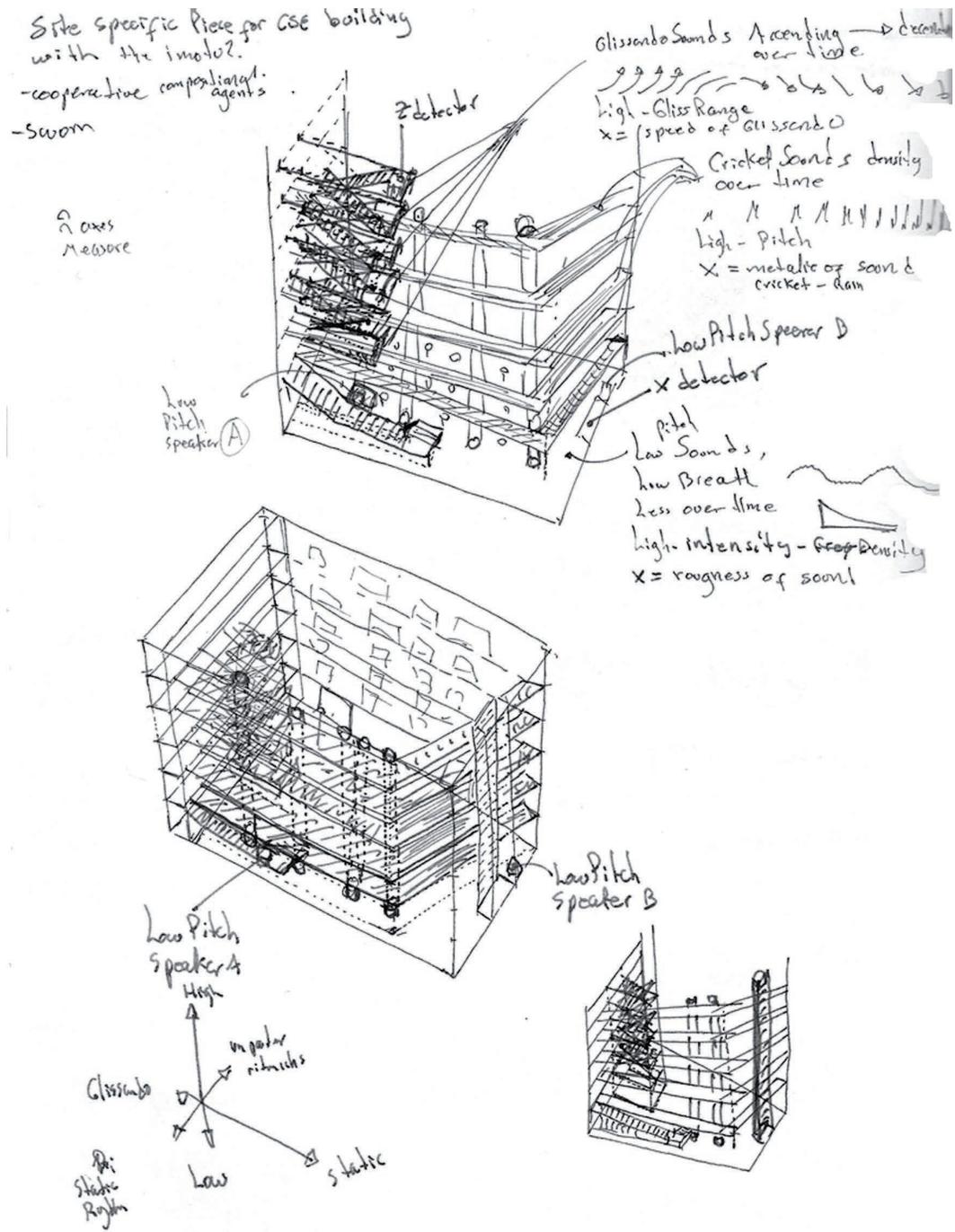


Figura 9.1. | Boceto de la distribución de superbirds para la pieza Swarms.

Para este proyecto, tales dispositivos electrónicos emulan criaturas orgánicas que se comunican a través de la radio estableciendo una red virtual a lo largo de la galería. Todas las criaturas están programadas con el mismo conjunto de rutinas básicas, pero su comportamiento sónico y visual en un momento dado depende de la interacción entre sus vecinos más cercanos. De manera similar a las criaturas orgánicas, estos agentes digitales transitan entre diferentes estados como la producción de gestos sonoros y visuales, escuchando y copiando otros materiales, durmiendo y confrontando a otros agentes.

Bruce Hemingway, Brian Mayton, Hugo Solís y los estudiantes del curso *Software for Embedded Systems* en la Universidad de Washington hemos desarrollado este trabajo.

Bruce Hemingway y su equipo de la Universidad de Washington desarrollaron el tablero *superbird* durante los últimos años. Su conjunto de capacidades como la síntesis de audio, la pantalla a color y la comunicación por radio las convierte en una tecnología adecuada para explorar experiencias audiovisuales interactivas. La placa contiene un microordenador *iMote* que ejecuta Linux. El dispositivo tiene un puerto de audio y es capaz de síntesis de audio en tiempo real.

Una veintena de estos dispositivos (cada uno conectado a un pequeño altavoz) se extiende a lo largo de la galería colgando de las paredes y del techo. Todos están conectados por radio creando una red. Mientras los dispositivos recrean el comportamiento de una colonia de enjambre el público escucha el resultado acústico de la evolución temporal de la interacción. Dado que los dispositivos miden la señal de la fuerza de los vecinos más cercanos el movimiento de la gente en la galería también influye en el resultado sónico.

ENSAYO DE JILL HARDY

Popularizada por la teoría de la información, la cibernética y la cultura informática, la idea es que todo el mundo puede ser tratado como datos listos para seleccionar, categorizar, visualizar y volver a visualizar con infinitas posibilidades. Hugo Solís y Bruce Hemingway trazan lo que no se ve al exponer el comportamiento de comunicación de los sistemas colectivos. *Swarms* es una red virtual que modela un sistema en el cual el comportamiento colectivo de un grupo es más sofisticado que el de sus miembros individuales, como las colonias de insectos, las escue-

las de peces y las bandadas de aves. Aproximadamente una docena de *superbirds* hechos de circuitos electrónicos están esparcidos por toda la galería, cada uno programado para crear sus propios sonidos parecidos a insectos, proyectados pensando en pequeños altavoces y parpadeando patrones luminosos exhibidos en pantallas LCD. Los *superbirds* detectan de forma inalámbrica las transmisiones visuales y sonoras de sus vecinos y se correlacionan internamente entre sí en una red virtual en toda la galería. Todos los *superbirds* están enviando y recibiendo información visual y de audio simultáneamente, y no hay líder ni jerarquía. Dependiendo de la proximidad, la intensidad de la señal y un elemento de aleatoriedad, el sonido y la luz evolucionan orgánicamente con el tiempo a medida que la configuración de la red se vuelve más sofisticada. Los visitantes de la galería también pueden recoger y reubicar los *superbirds*, interrumpiendo la red y pidiéndoles que reasignen a través del sonido y la luz. Incluso los visitantes de la galería del pensamiento pueden transformar el comportamiento de los enjambres, este elemento es menos importante que las interacciones de las aves como un colectivo. Solo y sin vecinos un pájaro permanecería estático y silencioso, pero como miembro de una comunidad, la variación y la complejidad surgen con el tiempo. *Swarms* traza un fragmento del mundo natural en constante evolución a través de composiciones audiovisuales que hacen visibles el comportamiento interno, que tradicionalmente es invisible. Publicado en ocasión de la exposición del Centro de Artes Kirkland Off the Map comisariada por Jill Hardy. ■ ■ ■



Metáforas para pianos muertos

Hoja de sala

Fecha de producción: 2009. *Lugares de exhibición:* JackStraw Gellery, Seattle; Bumbershot Seattle Festival, Guadalajara, Biblioteca Vansconcelos CDMX, Galería Manuel Felguerez del Centro Multimedia, Centro Nacional de las Artes CDMX. *Material:* Almas de pianos, componentes electrónicos, componentes mecánicos, sensores y microcomputadora. *Tecnologías creativas:* Código para SuperCollider y Java.

RESUMEN

Metáforas para pianos muertos es una serie de instalaciones sonoras interactivas que utilizan pianos antiguos que son considerados basura por sus antiguos dueños y son desmembrados e intervenidos. El objetivo poético es revivir los instrumentos utilizando una perspectiva sonora contemporánea. Los pianos son extendidos con circuitos electrónicos, microcontroladores, sensores, motores y solenoides. Este proyecto continúa las propuestas artísticas que a lo largo de la historia se han realizado en relación con el instrumento, entre las cuales podemos encontrar las obras de Annea Lockwood, *Piano Garden* (1970), *Piano Buring*, *Southern Exposure* y *Piano Transplant No. 4* (2005); *Two Suspended Grand Pianos* (1990) de Paul Panhuysen; y *Long Aeolian Piano* (1984 -1986) de Gordon Monahan and Thaddeus Holowina (Licht, 2007).

Utilizando los sensores una computadora registra y analiza la actividad de la audiencia y genera los materiales que los pianos reproducen con los solenoides y los motores. El resultado sonoro es una modulación entre la interacción de los participantes y el espectro sonoro de *El clave bien temperado* de J. S. Bach.

DESCRIPCIÓN

Metáforas para pianos muertos es una colección de instalaciones sonoras construidas a partir de pianos acústicos que han sido desechados. Los instrumentos son desmembrados e intervenidos con motores y solenoides, y controlados con circuitos electrónicos y una computadora. La intención conceptual es reencarnar los pianos muertos en nuevos objetos sonoros.

La actividad de la audiencia en el espacio, registrada con cámaras de video y sensores de movimiento y luz, modula la producción sonora en tiempo real extraída del espectro sonoro de la pieza *El clave bien temperado* de J. S. Bach. De esta manera la obra busca ser una reencarnación en el sentido material, sonoro y conceptual de la acústica de tiempos pasados y remotos reinterpretada en el presente con conceptos y tecnologías contemporáneas.

La instalación consta de varios pianos antiguos acústicos alterados mediante procesos mecánicos y extendidos con sensores. Los pianos están interconectados de forma inalámbrica y controlados desde un módulo central inteligente. En su conjunto la instalación es un espacio para la creación colectiva por parte de una audiencia a la que se le invita a interactuar con su presencia para generar un material que es recopilado, analizado y transformado paulatinamente a lo largo de la permanencia de la instalación.

Metáforas para pianos muertos es un proyecto que continúa una línea creativa que el artista ha desarrollado a lo largo de los últimos años y que empezó como ejecutante de improvisaciones electroacústicas en piano. En el aspecto tecnológico esta propuesta continúa la exploración iniciada en la obra *Tell-Tale Piano*, descrita en el capítulo 8, misma que también se realizó modificando un piano antiguo.

Esta obra busca ser un trabajo sonoro el cual mediante tecnologías digitales, métodos de inteligencia artificial y canales contemporáneos de información ofrezca una lectura personal

al significado del pianoforte —como objeto sonoro— desde una perspectiva contemporánea en relación con la actualidad estética y acústica.

CONCEPTO

En el año 1700 Bartolomeo Cristofori inventó el pianoforte (Reblitz, 1993). Su aceptación por parte de los compositores y de la audiencia no fue inmediata; sin embargo, con el paso de los años, el instrumento se refinó y se convirtió en uno de los más representativos de la música occidental. Su repertorio es vasto y desempeña un papel importante en la dotación instrumental de diferentes estilos musicales.

Con la comercialización de los teclados electrónicos durante los años ochenta y noventa se especuló fuertemente sobre el ocaso del instrumento. Ahora sabemos que ni su calidad tímbrica ni su significado semiótico son sustituibles fácilmente. El pianoforte —o piano— es un instrumento musical con propiedades sonoras muy características, un objeto con una carga histórica y una trascendencia como signo y símbolo que lo mantienen indiferente a la modernidad digital. El piano, como objeto material y como objeto sonoro, aún pertenece vigente al imaginario colectivo, al imaginario de los creadores y al imaginario de su audiencia.

A lo anterior es importante agregar que las realidades acústica y estética de nuestros días son radicalmente diferentes de las de hace 300 años. Se han expandido las posibilidades tímbricas y estéticas hasta horizontes que muchas veces nos parecen ya invisibles e inalcanzables. En nivel musical los cambios han sido numerosos y en ocasiones diametralmente opuestos. ¿Cómo abordan los compositores la realidad del piano en nuestros días? ¿Cómo observa y escucha un espectador el instrumento? ¿Qué significado, qué impacto y qué trascendencia tiene interpretar una sonata clásica en el instrumento en estos momentos de la historia? ¿Cuántos pianos preparados al estilo de Cage faltan por descubrirse e inventarse?

La manera en la que se aborda la relación entre el piano y la modernidad es personal. En esta instalación se propone una lectura propia en cuanto a los elementos estéticos, que su vez es genérica en su intención por relacionar al piano como objeto con un entorno cada vez más dinámico, más volátil, más ubicuo y mucho más virtual.

DESARROLLO Y VERSIONES

La primera iteración de esta obra se presentó en septiembre de 2009; en noviembre de ese mismo año se presentó una nueva versión, una tercera en marzo de 2010 y la cuarta y última versión en agosto de ese mismo año. Durante esos meses de trabajo realicé cambios significativos en la generación de sonido y en los mecanismos de interacción. Es interesante hacer una revisión de los procesos de transformación que sufrió la propuesta desde su planteamiento inicial, antes de iniciar la producción de la obra, hasta las últimas versiones más decantadas y depuradas cuando ya se tenía una visión sonora y un marco tecnológico

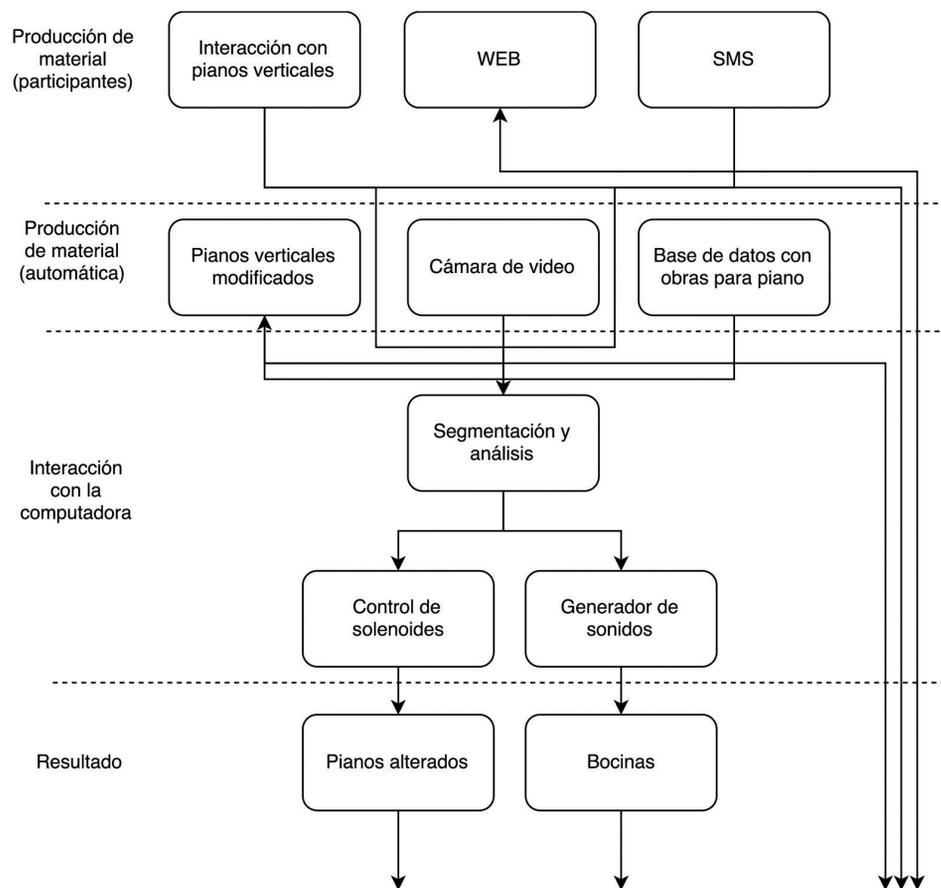


Figura 10.1. | Diagrama de la propuesta *Metáfora performativa del pianoforte*.

desarrollado. Habrá que poner atención en la innecesaria complejidad de la propuesta inicial y cómo durante su implementación se va convirtiendo, por razones estéticas, en un trabajo más sencillo con menos elementos tecnológicos.

Propuesta de obra

La propuesta inicial enunciaba de manera abstracta lo siguiente:

La instalación consta de cuatro pianos verticales antiguos alterados con procesos mecánicos, una cámara de video y una página web. En su conjunto la instalación sería un espacio para la creación espontánea por parte de una audiencia a quien se le invita a interactuar local y remotamente –con una página web– para generar un material que será recopilado, analizado y transformado paulatinamente a lo largo de la existencia de la instalación.

Metáfora performativa del pianoforte es una instalación técnicamente compleja. Está compuesta de varios módulos interconectados entre sí, la mayoría almacenando, transmitiendo, analizando y generando información en tiempo real. El material final y su apreciación son el resultado de la interacción entre los diferentes procesos digitales y su relación con el público y los ejecutantes.

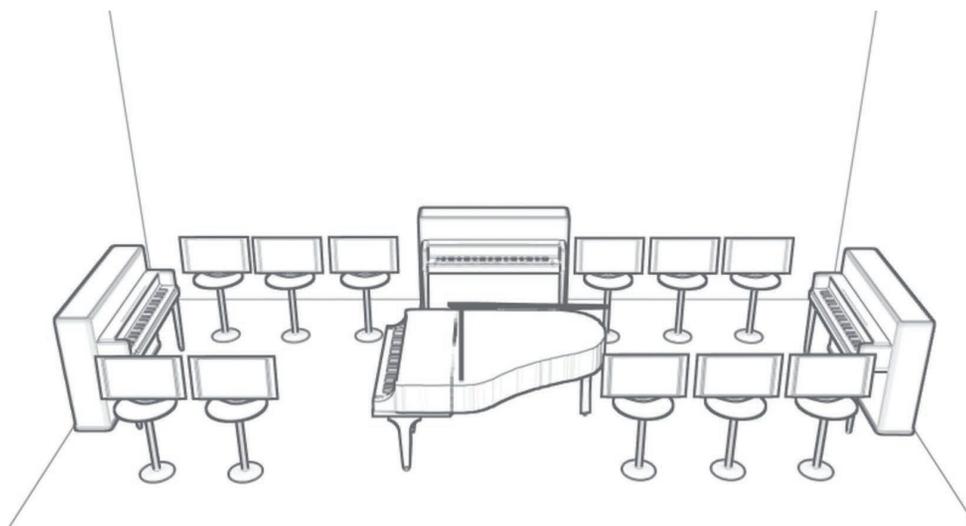


Figura 10.2. | Boceto digital de la propuesta *Metáfora performativa del pianoforte* (primera idea).

La instalación puede dividirse en cuatro partes de acuerdo con su función:

- Producción del material por parte de los participantes.
- Producción automática del material.
- Segmentación, análisis y generación de contenido.
- Presentación de contenido.

Producción del material por parte de los participantes

Una parte del contenido es derivado del material que se genera por parte de una audiencia activa. Todo el material generado por los espectadores es almacenado desde el montaje de la instalación con lo cual el contenido se va enriqueciendo a lo largo de los días. De esta manera se genera una historia de la actividad de la instalación donde la audiencia puede contribuir de diferentes maneras:

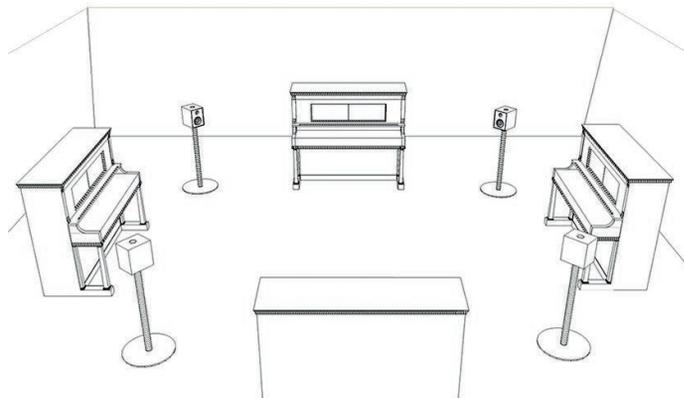


Figura 10.3. | Boceto digital de la propuesta *Metáfora performativa del pianoforte* (segunda idea).

Interactuando en el sitio web de la instalación En una página web diseñada ex profeso para la instalación, la audiencia puede observar el material generado en tiempo real y puede (subir) contribuir con nuevo material para su inclusión en la base de datos que se usa a fin de generar el contenido. El audio es procesado con el resto del material acústico generado por los pianos verticales y por el audio de la base de datos con música para piano.

Enviando mensajes SMS desde su teléfono celular La audiencia es invitada a enviar textos desde su teléfono celular. Si el texto enviado pertenece a una lista de textos previamente definidos entre los que se incluyen términos musicales que aparecen comúnmente en las partituras musicales, entonces el texto tiene una influencia directa en la generación del material. Por ejemplo, si una persona manda las palabras “presto fortissimo” entonces el material acústico generado por la instalación aumentará en intensidad, y se incrementará la cantidad de eventos gráficos y la velocidad de transición entre ellos.

Producción automática del material

Además de la generación directa del material por parte de los participantes la instalación tiene una segunda capa para producir materiales fuente. La producción “automática” de material también está influenciada por el material producido por los participantes; sin embargo, existe una diferencia sustancial entre la generación directa del material y la generación automática resultado de la actividad de los participantes. En esta segunda categoría encontramos:

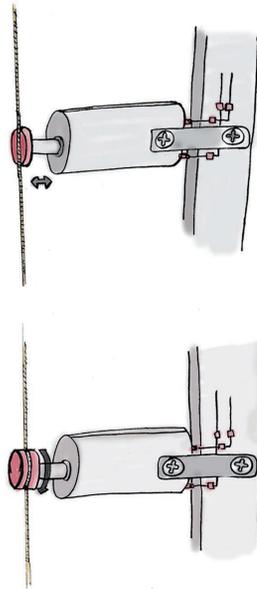


Figura 10.4. | Boceto de motor y solenoide.

Material acústico de los pianos verticales Los pianos verticales son intervenidos con pequeños circuitos mecánico-electrónicos diseñados específicamente para la instalación. Cada circuito tiene una entrada para comunicarse con la computadora y una o varias salidas para controlar pequeños solenoides, motores rotatorios y bobinas. Los solenoides se utilizan para golpear las cuerdas de los pianos en diferentes registros. Las partes móviles de los solenoides pueden ser cubiertas con diferentes materiales de densidades diversas —desde metales hasta fieltros afelpados— lo que produce tímbricas diferentes. Los motores rotatorios se pueden utilizar para frotar las cuerdas con materiales rugosos y semirrugosos. Con las funciones PWM (Pulse Width Modulation) del microcontrolador es posible hacer girar los motores a diferentes velocidades y con ello obtener sonoridades variadas. Las bobinas pueden ser utilizadas como electroimanes para hacer oscilar por empatía las cuerdas de los pianos. La intervención se realiza directamente dentro del alma del piano por lo que el resultado sonoro no es muy intenso, ya que se trata de pequeños sensores piezoeléctricos que son utilizados como micrófonos de contacto. La señal preamplificada es enviada a la computadora para su procesamiento y su posterior envío a las bocinas.

Base de datos con música para piano Una base de datos personal con aproximadamente 150 horas de música para piano de diferentes periodos y estilos es el punto acústico de partida de la instalación. Este material está codificado y etiquetado correctamente en formato mp3 y no se utiliza en su forma original, sino que primero se somete a procesos de segmentación y clasificación. Una vez segmentado y clasificado los diferentes “motivos” son transformados en tiempo real de acuerdo con la conducta y la interacción de los participantes.

Cámara de video La instalación cuenta con una cámara de video colocada en la parte central del techo que cubre toda la escena. La cámara registra el movimiento de las personas alrededor de la sala. Las imágenes no se utilizan de manera directa, sino que de ellas se extrae información. Un ejemplo de lo anterior es ubicar la cantidad y la ubicación de las personas en la sala.

Interacción con la computadora

Es importante mencionar que el material producido por la computadora se obtiene de segmentar, analizar, clasificar y transformar el contenido generado por los participantes. Uno de

los principales problemas que tiene el arte multimedia de nuestros días es la simplicidad con la que se maneja su contenido y el “mapeo” entre sus partes. Una solución parcial a este complejo problema es el empleo de técnicas que pertenecen a las áreas de la inteligencia artificial, el aprendizaje automático (*machine learning*), la minería de datos y la recuperación de información. El empleo de las técnicas de dichas áreas no garantiza un resultado estético interesante, pero sí puede contribuir a realizar instalaciones con interacciones de más alto nivel.

Entre los métodos para segmentar y clasificar el contenido se exploran los árboles de decisión, las máquinas de soporte vectorial y otros procesos de aprendizaje supervisado y no supervisado dependiendo de las características del contenido. Para las transformaciones se exploran las cadenas de Markov normales y escondidas, los algoritmos genéticos, las redes bayesianas, los procesos gaussianos y las redes neuronales, entre otros.¹ Los materiales originales producidos por el público, por los procesos automáticos y los previamente almacenados son manipulados de acuerdo con la conducta del público activo. En ningún caso estos materiales son reproducidos en su forma original, sino que son transformados y homogeneizados con criterios estéticos muy definidos.

Manipulación de los gráficos En el caso del video para analizar la actividad del público, la señal es sometida a las técnicas de reconocimiento de objetos. Ninguna es cien por ciento fiable pero bajo condiciones controladas es posible cuantificar de forma aproximada el número de personas en la sala, su ubicación y su velocidad de desplazamiento.

Manipulación del audio En muy pocos casos el sonido de las fuentes originales es reproducido sin transformación. Tanto el material generado por la audiencia como el material generado por los pianos verticales y la música para piano son constantemente procesados por la computadora. Reverberaciones, cambios radicales en la velocidad de reproducción, cambios de frecuencia y síntesis granular son algunos de los procesos que se aplican. El audio resultante es distribuido en las bocinas con lo cual se consiguen algunos efectos de especialización sonora.

¹ Estas técnicas de análisis y generación de datos son procedimientos tradicionales en el campo de las ciencias de la computación. Son las bases fundamentales de lo que conocemos como *inteligencia artificial*. A las personas interesadas se les invita a revisar cualquier libro de consulta relacionado con el tema, como por ejemplo, Russell y Norving (1995).

Tecnología utilizada

El sistema es controlado con el software Juum. Juum es un híbrido entre Processing, SuperCollider, MaxMSP y Fluid. Juum tiene una parte central en Java, una sección para escribir scripts en Groovy, una serie de librerías en C para el control de Gráficos OpenGL (OpenGL) y comunicación OSC para interactuar con SuperCollider.

Para los módulos que se desarrollan más fácil y rápidamente en lenguajes script se utiliza Groovy o Jython por su imbricación con Java y su agilidad en escritura. Para las librerías y módulos que requieren optimización se utiliza C/C++.

Interconexión La instalación consta de numerosos módulos y diferentes tecnologías. Para poder comunicarse entre sí se utiliza el protocolo Open Sound Control (OSC). Numerosas implementaciones de código abierto de este protocolo están disponibles en internet.

Análisis del audio Analizar el contenido musical de archivos de audio digital es un trabajo complejo que no era posible hasta hace pocos años y que sigue siendo área de investigación. En la actualidad hay ya varias librerías especializadas en extraer información musical de archivos de audio con el fin de segmentar, clasificar y reconstruir bases de datos musicales. Una de ellas es el proyecto Meapsoft que permite segmentar el audio en pulsos o eventos sonoros.² Una vez segmentados, cada evento de audio es analizado y se extraen de él las siguientes propiedades:

Vector de distribución de la energía en cada semitono de la escala.

- Semitono dominante.
- Energía promedio del segmento.
- Estimación de la frecuencia del segmento.
- Promedio del espectro convertido y ponderado con una escala Mel.
- Promedio MFCC. Descriptor utilizado comúnmente en reconocimiento del habla.
- Estimación del tono.
- Promedio del centro de masa espectral.
- Estimación de la cantidad de picos en el segmento.

² Al momento en que se escribió este texto en su versión original para una convocatoria en 2008, Meapsoft era una de las librerías con mejor documentación. Al día de hoy, en 2018, Librosa es una librería para obtener descriptores de audio mucho más actualizada.

- Promedio del espectro.
- Duración del segmento.
- Estabilidad espectral.

Estas propiedades nos permiten clasificar, reordenar y alterar tanto la base de datos con el material grabado, como *i*) el material que los asistentes generen en el piano, *ii*) el material generado en los pianos verticales, y *iii*) el material generado a través de la página web.

Análisis de la imagen Para registrar el material detectado por la cámara de video se utiliza el Java Media Framework API (JMF). Esta librería es una extensión de Java y, por tanto, es sencillo de vincular con el resto del proyecto. Instalaciones previas ya han sido creadas con esta librería que, además, permite implementar la interacción multimedia en la página web.

sms Para registrar los mensajes telefónicos que manda la audiencia se utiliza la librería SMSLib. Esta librería permite mandar y recibir mensajes SMS con una computadora conectada a un teléfono con modem GSM. La librería funciona en Java, en .NET, es gratuita y de código abierto. (Nota: la he utilizado en trabajos anteriores.) *WEB* Para la elaboración de la página web se utiliza un servidor Apache y Drupal para crear el contenido. Drupal es un administrador de contenido gratuito que permite crear, entre otras cosas, blogs, usuarios, galerías, eventos y noticias de manera muy sencilla. El contenido es muy fácil de editar y ordenar y su presentación es ligera y elegante.

Intervención de los pianos verticales Cada circuito es controlado mediante un microcontrolador ATmega168 que es el mismo que se encuentra en las tarjetas Arduino. El puerto serial de la computadora principal se conecta a un multiplexor y éste se utiliza para controlar los diferentes circuitos. Cada microcontrolador cuenta con 12 puertos de entrada/salida por lo que dos o tres circuitos por piano son suficientes para generar sonidos en registros diferentes y con calidades tímbricas variadas.

Generación de audio El programa principal para la manipulación y el procesamiento del material pregrabado en la base de datos y el material producido en los pianos verticales es Supercollider. Supercollider es un lenguaje de programación diseñado particularmente para la síntesis de sonido y es utilizado frecuentemente en el campo de la música por computadora.

Administración de la información Toda la información almacenada y generada antes y durante la presentación de la instalación, el audio, los gráficos y la actividad de los participantes es almacenada en una base de datos. Se utilizará la base de datos Hypersonic Database System (HSQLDB).

EVOLUCIÓN EN EL DESARROLLO DE LA OBRA

Primer prototipo



Figura 10.5. | Arreglo de solenoides sobre base de aluminio.

En la sección anterior se presentó una propuesta detallada de la obra como descripción de concepto antes de iniciar la producción. Es interesante observar que durante el proceso de desarrollo muchos de los elementos propuestos se descartaron y otros adquirieron mayor fuerza de la que se tenía contemplada inicialmente.

El primer prototipo estuvo enfocado al control de motores DC que utilizaban feltros para frotar las cuerdas y solenoides pequeños para activarlas. Estos experimentos no fueron del todo satisfactorios y después de producir bases tanto de plástico como de aluminio en máquinas CNC y luego de hacer las tarjetas de potencia con arreglos de transistores me percaté

de que las partes mecánicas se gastaban a una velocidad tal que era imposible tenerlas en exhibición por el constante mantenimiento requerido.

Durante esta etapa del proceso decidí utilizar una microcomputadora en lugar de una computadora para el control de los datos y descarté la idea de permitir al público la interacción directa con el objeto. En lugar de utilizar cámaras para el rastreo decidí hacer una matriz de sensores lumínicos que permitieran rastrear las sombras de los espectadores. Para ello fue necesario colocar una luz infraroja que aumentara la diferencia de luminosidad con la presencia o ausencia de espectadores.

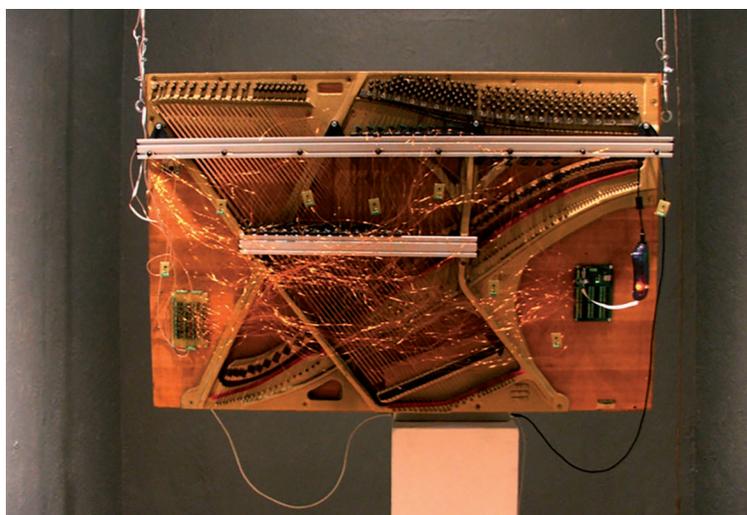


Figura 10.6. Pruebas de montaje con todo el sistema de control.

Primera versión

La experiencia adquirida durante el primer prototipo me llevó al diseño de una versión en la que continué con el rastreo de los espectadores a través del cambio de luz, pero que ahora utilizaba los motores como objetos percutores semicaóticos que ya no estaban fijos en bases, sino que colgaban libremente del cable de alimentación. Cada motor tiene un percutor de piano que desplaza su peso y con ello lo hace brincar y golpear las cuerdas sin la necesidad de utilizar mucha potencia o de activarlos de manera constante. Para esta versión utilicé ya los cuatro pianos propuestos inicialmente.

En la figura 10.7 se observan los cuatro pianos suspendidos desde el techo a una altura aproximada de dos metros del nivel del piso. En cada uno de ellos cuelgan los sensores de luz que apuntan al suelo y una serie de motores de baja potencia que al activarse golpean diferentes partes del piano pudiendo ser cuerdas diversas o la caja de resonancia directamente.



Figura 10.7. | *Metáforas para pianos muertos* en la Biblioteca Vasconcelos CDMX. Fotografía de Nirvana Paz.



Figura 10.8. | Detalle de *Metáforas para pianos muertos*, versión de la Biblioteca Vasconcelos CDMX.

Segunda versión

La siguiente versión de la propuesta llevó al proyecto a una reducción aún mayor de la parte mecánica. Aunque el sistema ofrecía una paleta sonora cada vez más rica, el deseo de exploración me condujo a experimentar con pequeños motores vibratorios de teléfonos celulares. Éstos se colocaron en diferentes partes de las almas de los pianos utilizando ahora silicón, lo que les daría juego y libertad a los motores. El consumo de corriente es mínimo y el mantenimiento es menor. Un elemento estético importante en esta versión es el uso de cable de bobina para todo el sistema. El cable de bobina es de cobre al igual que el entorchado de las cuerdas del piano y así se refuerza la idea de que el sistema electrónico es una extensión del organismo del piano.

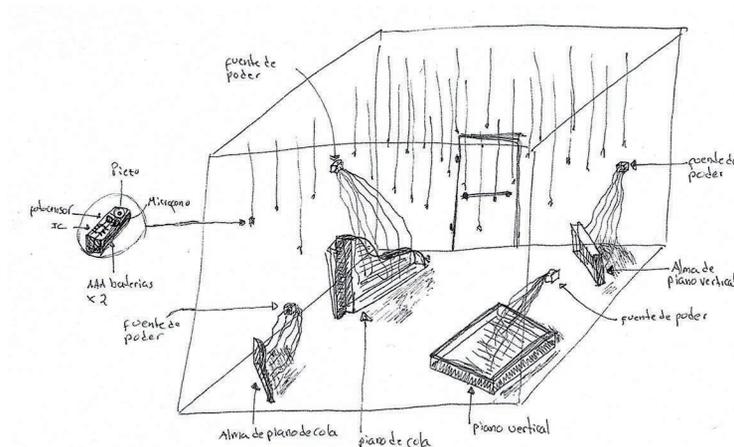


Figura 10.9. | Boceto de la segunda versión de la obra *Metáforas para pianos muertos*.

La pieza fue exhibida en varias ocasiones, pero se mantuvo la constante de hacer la matriz de detección de luminosidad en todo el espacio de exhibición con la intención de apropiarse del mismo y poder hacer una detección más amplia del territorio donde se encontraban los espectadores. Para esta versión se regresó al uso de una computadora personal para realizar las operación de rastreo de los espectadores.



Figura 10.10. | Espectadora interactuando con la pieza *Metáforas para pianos muertos*.

RELACIÓN DATOS-EXPECTADORES

Análisis de la obra de Bach

Una de las preocupaciones principales a lo largo de este proyecto fue el concepto de *reinterpretación*. En el sentido físico se reinterpretó el instrumento musical reutilizando el dispositivo material. En el sentido acústico se buscó la reinterpretación de la música para teclado y para ello utilicé una obra icónica. *El clave bien temperado* de J. S. Bach es una obra importante en la historia de la música, no sólo por su calidad musical, sino por haber sido una prueba musical de los beneficios que ofrecía la afinación temperada. Esta obra es, por tanto, un símbolo que contiene información intrínseca y externa a su contenido musical.

Lo primero que hice fue extraer información sobre las propiedades de *El clave bien temperado* de Bach. Para ello utilicé un programa de extracción de descriptores de audio desarrollado por Tristan Jehan (2005) que permite segmentar archivos de audio digital. Cada audio es segmentado en eventos perceptualmente diferenciables, ya sea por cambios de intensidad o de timbre. El analizador permite obtener información de diferentes parámetros tímbricos para cada segmento.³ La información de cada archivo de audio se regis-

³ Este motor de análisis musical se utilizó dentro de la empresa EchoNest la cual se dedicó a clasificar y recomendar música antes de ser adquirida por Spotify. Al día de hoy sería mucho mejor utilizar otra librería como podría ser Librosa.

tra en un archivo de formato XML. De esta manera, el programa recibe un archivo de audio y genera un XML con información sobre las propiedades tímbricas del mismo.

El siguiente es un fragmento de código que muestra el tipo de información que se genera al analizar un archivo de audio:

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <analysis decoder="Quicktime" version="0x7608000">
3.   <track duration="160.00000" endOfFadeIn="0.00000"
4.     startOfFadeOut="146.62240" loudness="-15.413"
5.     tempo="86.593" tempoConfidence="0.720"
6.     timeSignature="4" timeSignatureConfidence="0.748"
7.     key="5" keyConfidence="0.478"
8.     mode="1" modeConfidence="1.000">
9.   <meter>
10.    <bar conf="-1">
11.     <beat conf="0.936">
12.      <tatum conf="1">0.74923</tatum>
13.      <tatum conf="0.634">1.09898</tatum>
14.    </beat>
15.    <beat conf="0.770">
16.     <tatum conf="0.764">1.44822</tatum>
17.     <tatum conf="0.487">1.79856</tatum>
18.    </beat>
19.    <beat conf="0.753">
20.     <tatum conf="0.728">2.14896</tatum>
21.     <tatum conf="0.515">2.50119</tatum>
22.    </beat>
23.  </bar>
24. </meter>
25. <sections>
26.   <section start="0.00000" duration="8.56942"/>
27.   <section start="8.56942" duration="6.59508"/>
28.   <section start="15.16450" duration="11.02938"/>
29.   <section start="26.19388" duration="10.54596"/>
30.   <section start="36.73984" duration="23.78161"/>
31.   <section start="60.52145" duration="9.92457"/>
32.   <section start="70.44602" duration="9.33994"/>
33.   <section start="79.78596" duration="12.57006"/>
34.   <section start="92.35603" duration="23.13264"/>
35.   <section start="115.48867" duration="8.18364"/>
36.   <section start="123.67231" duration="11.76120"/>
37.   <section start="135.43351" duration="13.86031"/>
38.   <section start="149.29382" duration="10.70618"/>
39. </sections>
40. <segments>
41.   <segment start="0.00000" duration="0.68757">
42.     <loudness>
43.       <dB time="0">-51.063</dB>

```

```

44.     <dB time="0.04644" type="max">-49.661</dB>
45. </loudness>
46. <pitches>
47.   <pitch class="0">0.844</pitch>
48.   <pitch class="1">0.621</pitch>
49.   <pitch class="2">1.000</pitch>
50.   <pitch class="3">0.993</pitch>
51.   <pitch class="4">0.679</pitch>
52.   <pitch class="5">0.541</pitch>
53.   <pitch class="6">0.347</pitch>
54.   <pitch class="7">0.474</pitch>
55.   <pitch class="8">0.577</pitch>
56.   <pitch class="9">0.477</pitch>
57.   <pitch class="10">0.337</pitch>
58.   <pitch class="11">0.326</pitch>
59. </pitches>
60. <timbre>
61.   <coeff dim="0">7.801</coeff>
62.   <coeff dim="1">49.658</coeff>
63.   <coeff dim="2">-92.419</coeff>
64.   <coeff dim="3">-33.390</coeff>
65.   <coeff dim="4">-0.101</coeff>
66.   <coeff dim="5">-55.570</coeff>
67.   <coeff dim="6">22.488</coeff>
68.   <coeff dim="7">38.494</coeff>
69.   <coeff dim="8">-3.247</coeff>
70.   <coeff dim="9">-2.193</coeff>
71.   <coeff dim="10">-17.593</coeff>
72.   <coeff dim="11">-14.293</coeff>
73. </timbre>
74. </segment>
75. </segments>
76. </track>
77. </analysis>

```

Como se puede observar en el código, el audio es analizado y se extraen algunos descriptores generales como tempo y tonalidad. Se extraen los compases y los segmentos. De cada segmento se extrae una intensidad, un cromatismo y un descriptor de 11 dimensiones que representa la propiedad tímbrica del segmento.

Esta información también puede ser visualizada gráficamente. Como se puede ver en la figura 10.11 la información nos permite ver estructuras tímbricas, tonalidades y modulaciones, así como estructuras de intensidad y de variación en velocidad en los segmentos.

Realicé los scripts en el lenguaje *Groovy* necesario para analizar todas las piezas que componen *El clave bien temperado* y la información fue grabada tanto en bases de datos como en

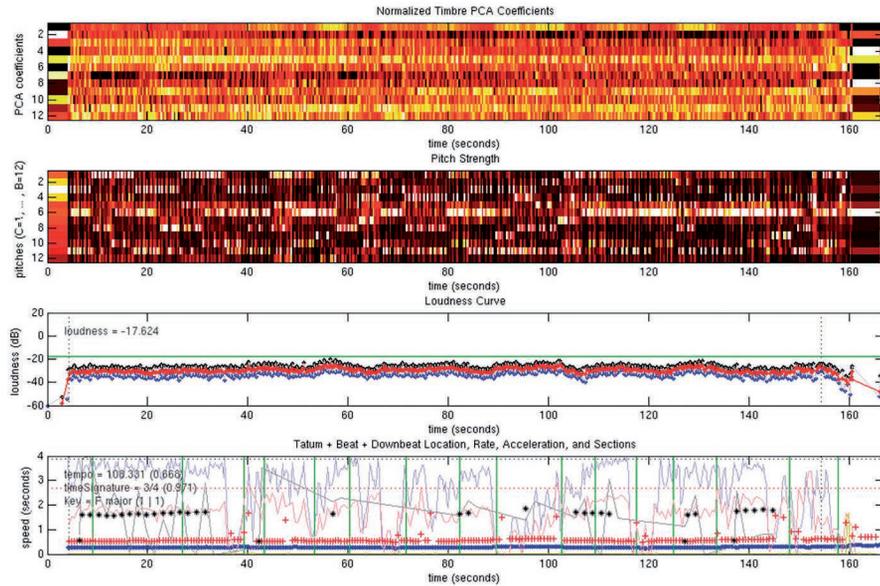


Figura 10.11. | Visualización de la información del análisis de audio.

archivos de texto. La información en archivos de texto permitía guardarla en la microcomputadora sin necesidad de generar una base de datos.

```

1. import groovy.sql.Sql
2. //data base creation and schema creation
3. db = Sql.newInstance('jdbc:hsqldb:file:bach','sa',
4.    '','org.hsqldb.jdbcDriver')
5. piecesSet = db.dataSet('Pieces')
6. segmentsSet = db.dataSet('Segments')
7. segContentSet = db.dataSet('SegsContent')
8. piecesSegmentsSet = db.dataSet('PieceSegments')
9. piecesSegmentsContentSet = db.dataSet('PieceSegmentsContent')
10.
11. data = []
12. record = piecesSegmentsContentSet.findAll{it.fileName
13.    == '1 In C, Bwv 846 Fugue.aif.xml'}
14. piecesSegmentsContentSet.each{
15.    data << [it.dB,
16.        it.pitch_00,it.pitch_01,it.pitch_02,
17.        it.pitch_03,it.pitch_04,it.pitch_05,
18.        it.pitch_06,it.pitch_07,it.pitch_08,
19.        it.pitch_09,it.pitch_10,it.pitch_11,
20.        it.timbre_00,it.timbre_01,it.timbre_02,
21.        it.timbre_03,it.timbre_04,it.timbre_05,
22.        it.timbre_06,it.timbre_07,it.timbre_08,

```

```
23.         it.timbre_09,it.timbre_10,it.timbre_11]
24.     }
25.
26.     myFile = new File('bach.txt')
27.     myFile.write(data.size() + '\n')
28.     data.each{
29.         myFile << it.join(' ') + '\n'
30.     }

println 'done'
```

Correlación entre el desplazamiento del público y lo datos musicales

Diferentes tipos de *mapeos* entre la generación del audio y la actividad del público fueron diseñados y probados. Uno de los mecanismos que mejor funcionó a nivel estético, dado que daba pie a la sorpresa, a la curiosidad y, sobre todo, que no era obvio en su respuesta y preservaba la lógica de utilizar el espectro de *El clave bien temperado* fue un sistema en donde el público funcionaba como una compuerta para el flujo de información en cierto registro del instrumento.

La información tímbrica descrita en la sección anterior era leída en bucle en la computadora conservando el tempo de reproducción de la obra original. Los sensores de luz estaban colocados de forma tal que había sensores repartidos en todo el piano y, por tanto, estaban repartidos a lo largo de todo el registro del piano. La información de los valores de luz era recibida por los puertos convertidores analógicos digitales del *Plurifocal Event Controller* y enviados a la computadora. Si se excedía cierto umbral de valor se asumía que había una persona frente a dicho sensor y, por tanto, enfrente de una región particular del piano. En ese momento se activaban los motores de dicho registro con la intensidad que dictara en ese momento el espectro de *El Clave bien temperado*. De esta manera el participante era un modulador en el terreno espacial de *El Clave bien temperado*. Si revisamos con atención podemos ver que el público funcionaba como una compuerta lógica en la multiplicación de los datos del espectro. El espectro podía tener en un momento determinado valor alto de energía, pero los motores no se prendían si no había nadie enfrente de dicho registro en el piano. Por su parte, la simple presencia de la audiencia no garantizaba la activación de los motores si en ese momento no había energía en el espectro de la pieza.

Adicionalmente, en la versión que se presentó en la Biblioteca Vasconcelos el sistema no podía estar activo constantemente, pues el recinto necesitaba de grandes periodos de silencio y la pieza debía de activarse por pocos minutos cada hora por lo que el código tenía momentos de escucha y reposo.

Esta mecánica rompía con la linealidad y el mapeo directo y, al mismo tiempo, conservaba el diálogo entre las partes que constituían la pieza: el objeto físico, el material musical de *El Clave bien temperado* y el público participante.

A continuación se presenta el código que la microcomputadora utilizaba para leer los valores analógicos de los cuatro circuitos (uno para cada piano) e integrarlos y enviar nuevamente a los circuitos con las órdenes de cuáles motores debían de activarse y cuáles no. ■■■

```

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <stdint.h>
4. #include <fcntl.h>
5. #include <errno.h>
6. #include <termios.h>
7. #include <string.h>
8. #include <unistd.h>
9. #include <time.h>
10.
11. #define ESCUCHANDO_TIME 20
12. #define TOCANDO_TIME 120
13. #define DESCANSANDO_TIME 20
14. #define ESCUCHANDO 0
15. #define TOCANDO 1
16. #define DESCANSANDO 2
17. #define MAX_NUM_SOLENOIDS 4
18. #define UNIT_OF_SENSING 10
19.
20. int serial_fd;
21. unsigned int current_mode = ESCUCHANDO;
22.
23. void send_single(unsigned int solenoid, unsigned int value){
24.     char send_buf[128], data_buf[128];
25.     sprintf(data_buf, "ATD 55,1,8\rSN %02x %02x\r",
26.         solenoid, value);
27.     strcpy(send_buf, data_buf);
28.     write(serial_fd, send_buf, strlen(send_buf));
29.     printf("send_single %02x %02x\n", solenoid, value);
30. }
31.
32. void send_four(unsigned int value, unsigned int sol1,
33.     unsigned int sol2, unsigned int sol3, unsigned int sol4){

```

```

34. char send_buf[128], data_buf[128];
35. sprintf(data_buf, "ATD 55,1,17\rSS %02x %02x %02x %02x %02x\r",
36. value, sol1, sol2, sol3, sol4);
37. strcpy(send_buf, data_buf);
38. write(serial_fd, send_buf, strlen(send_buf));
39. printf("send_four %02x %02x %02x %02x\n",
40. value, sol1, sol2, sol3, sol4);
41. }
42.
43. void send_all_same(unsigned int value){
44. char send_buf[128], data_buf[128];
45. sprintf(data_buf, "ATD 55,1,5\rAS %02x\r", value);
46. strcpy(send_buf, data_buf);
47. write(serial_fd, send_buf, strlen(send_buf));
48. printf("send_all_same %02x\n",value);
49. }
50.
51. void send_globalRatio(unsigned int value){
52. char send_buf[128], data_buf[128];
53. sprintf(data_buf, "ATD 55,1,5\rGR %02x\r", value);
54. strcpy(send_buf, data_buf);
55. write(serial_fd, send_buf, strlen(send_buf));
56. printf("send_globalRatio %02x\n",value);
57. }
58.
59. void send_globalTempo(unsigned int value){
60. char send_buf[128], data_buf[128];
61. sprintf(data_buf, "ATD 55,1,5\rGT %02x\r", value);
62. strcpy(send_buf, data_buf);
63. write(serial_fd, send_buf, strlen(send_buf));
64. printf("send_globalTempo %02x\n",value);
65. }
66.
67. int main(int argc, char **argv) {
68. //variables for serial
69. char *port = "/dev/ttySAC1";
70. char buf[128];
71. int i, n;
72. char newdata[128];
73. char recv_buf[256];
74. char cmd_str[128];
75. char analogValuesStr[128];
76. unsigned int analogValues[8];
77. unsigned int previousValues[8];
78. uint8_t rdptr, wrptr;
79. struct termios tio;
80.
81. //variables for file handling
82. FILE *pfile = NULL;
83. char* fileName = "/sdcard/bach.txt";
84. signed int currentSegments[24];

```

```
85. unsigned int headInfo[1];
86. unsigned int currentLine = 0;
87.
88. //variables for control
89. unsigned int analogThreshold[8] = {20,20,20,20,20,20,20,20};
90. unsigned long previousSecond = 0;
91. unsigned int counter = 0;
92. unsigned int amountOfChange[8] = {0,0,0,0,0,0,0,0};
93.
94. unsigned int sumOfChange, numOfSolenoids, timeGap;
95.
96. //init of serial
97. serial_fd = open(port, O_RDWR | O_NONBLOCK | O_NOCTTY);
98. if(serial_fd < 0) {
99.     perror(port);
100.    exit(1);
101. }
102.
103. tcgetattr(serial_fd, &tio);
104. cfsetspeed(&tio, 38400);
105. tcsetattr(serial_fd, TCSANOW, &tio);
106.
107. rdptr = wrptr = 0;
108.
109. usleep(1000000);
110.
111. //init of networking
112. char send_buf[128];
113.
114. printf("\nLeaving if previous\n");
115. strcpy(send_buf, "AT+WLEAVE\r");
116. write(serial_fd, send_buf, strlen(send_buf));
117. usleep(1000000);
118.
119. printf("\nSetup zigbit\n");
120. strcpy(send_buf, "ATE0Q0V0X0\r");
121. write(serial_fd, send_buf, strlen(send_buf));
122. usleep(1000000);
123.
124. printf("\nSetup Network\n");
125. strcpy(send_buf,
126. "AT+GSM=1 +WPAID=1620 +WCHMASK=100000" +
127. "+WROLE=0 +WSRC=0 +WPASSWORD 0\r");
128. write(serial_fd, send_buf, strlen(send_buf));
129. usleep(1000000);
130.
131. printf("\nJoining\n");
132. strcpy(send_buf, "AT+WJOIN\r");
133. write(serial_fd, send_buf, strlen(send_buf));
134. usleep(1000000);
135.
```

```
136. //init of file handling
137. pfile = fopen(fileName, "r");
138. fscanf(pfile, "%d", &headInfo[0]);
139.
140. send_all_same(0x00); //just for turning off remaining
141. //usleep(1000000);
142. //send_globalRatio(200);
143.
144. while(1) {
145.
146. ///control of solenoids
147. time_t currentSecond;
148. currentSecond = time (NULL);
149. srand((unsigned int)currentSecond);
150.
151. if(currentSecond > previousSecond + 1){
152. //inside here everything happens every second
153.
154. if (current_mode == ESCUCHANDO) {
155. //hacer escucha
156. if(counter > ESCUCHANDO_TIME){
157. printf("cambiando a tocar %d\n", counter);
158. printf("result of listening %d %d %d %d %d %d %d\n",
159. amountOfChange[0],
160. amountOfChange[1],
161. amountOfChange[2],
162. amountOfChange[3],
163. amountOfChange[4],
164. amountOfChange[5],
165. amountOfChange[6],
166. amountOfChange[7]);
167.
168. sumOfChange = 0;
169. for (i = 0; i < 8; i++) sumOfChange += amountOfChange[i];
170. numOfSolenoids = (int)(sumOfChange / UNIT_OF_SENSING);
171. if (numOfSolenoids > MAX_NUM_SOLENOIDS)
172. numOfSolenoids = MAX_NUM_SOLENOIDS;
173. if (numOfSolenoids > 0)
174. timeGap = (int)((TOCANDO_TIME - 10) / numOfSolenoids);
175.
176. printf(«result of listening is: sumOfChange = %d,
177. numOfSolenoids = %d, timeGap = %d\n»,
178. sumOfChange, numOfSolenoids, timeGap);
179.
180. counter = 0;
181. current_mode = TOCANDO;
182. }
183. }
184.
185. if (current_mode == TOCANDO) {
186. //hacer tocando
```

```

187. if(numOfSolenoids > 0 && counter % timeGap == 0){
188.     int myRandom = rand() % 32;
189.     send_single(myRandom, 0x01);
190. }
191. if(counter > TOCANDO_TIME){
192.     printf("cambiando a descansar %d\n", counter);
193.     for(i = 0; i < 8; i++)amountOfChange[i] = 0;
194.     counter = 0;
195.     current_mode = DESCANSANDO;
196. }
197. }
198.
199. if (current_mode == DESCANSANDO) {
200.     //hacer descansar
201.     //printf ("Descanzando %d\n", counter);
202.     if(counter % 5 == 0) send_all_same(0x00);
203.     if(counter > DESCANSANDO_TIME){
204.         send_globalRatio(200);
205.         printf("cambiand a escuchar %d\n", counter);
206.         counter = 0;
207.         current_mode = ESCUCHANDO;
208.     }
209. }
210.
211. counter++;
212. previousSecond = currentSecond;
213. }
214.
215. //SERIAL INPUT AND GETTING ANALOG VALUES
216. n = read(serial_fd, newdata, 128);
217. if(n > 0) {
218.     // we received some data (n bytes), it's in newdata
219.     // copy it into recv_buf (a circular buffer)
220.     for(i=0; i<n; i++) {
221.         recv_buf[wrptr] = newdata[i];
222.         //printf("%c", newdata[i]);
223.         fflush(stdout);
224.         wrptr++;
225.     }
226. }
227.
228. while(rdptr != wrptr) {
229.     strcpy(cmd_str, "8 0055,0,24:");
230.     if( wrptr - rdptr ) >= strlen(cmd_str) + 24) {
231.         if( (strncmp( &recv_buf[rdptr], cmd_str, strlen(cmd_str))
232. == 0) ) {
233.             rdptr = rdptr + strlen(cmd_str);
234.             for(i=0; i<24; i++) {
235.                 analogValuesStr[i] = recv_buf[rdptr];
236.                 rdptr++;
237.             }

```

```
238. analogValuesStr[i] = '\0';
239. //printf("\nGot analog values: %s\n", analogValuesStr);
240. sscanf(analogValuesStr, "%x %x %x %x %x %x %x %x",
241. &analogValues[0],
242. &analogValues[1],
243. &analogValues[2],
244. &analogValues[3],
245. &analogValues[4],
246. &analogValues[5],
247. &analogValues[6],
248. &analogValues[7]
249. );
250.
251. //READING LINE OF DATA FILE
252. if (currentLine >= headInfo[0]) {
253.     currentLine = 0;
254.     rewind(pfile);
255.     fscanf(pfile, "%d", &headInfo[0]);
256. }
257.
258.     fscanf(pfile, "%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f",
259.     &tSegments[0],
260.     &tSegments[1],
261.     &tSegments[2],
262.     &tSegments[3],
263.     &tSegments[4],
264.     &tSegments[5],
265.     &tSegments[6],
266.     &tSegments[7],
267.     &tSegments[8],
268.     &tSegments[9],
269.     &tSegments[10],
270.     &tSegments[11],
271.     &tSegments[12],
272.     &tSegments[13],
273.     &tSegments[14],
274.     &tSegments[15],
275.     &tSegments[16],
276.     &tSegments[17],
277.     &tSegments[18],
278.     &tSegments[19],
279.     &tSegments[20],
280.     &tSegments[21],
281.     &tSegments[22],
282.     &tSegments[23]);
283.
284.     currentLine++;
285.     //printf("*****GOT NEW DATA\n");
286.     for(i=0; i<8; i++) {
287.         if(analogValues[i] > previousValues[i] +
288.         analogThreshold[i] || analogValues[i] <
```

```
289.     previousValues[i] - analogThreshold[i]){
290.     //printf("Change in: %d\n", i);
291.     amountOfChange[i]++;
292.     }
293.     }
294.     //UPDATE PREVIOUS VALUES
295.     for(i=0; i<8; i++) {
296.     previousValues[i] = analogValues[i];
297.     }
298.     } else {
299.     rdptr++;
300.     }
301.     } else {
302.     break;
303.     }
304.     }
305.     }
306.     return 0;
307.     }
```




Objetos resonantes

Hoja de sala

Fecha de producción: 2009. *Lugares de exhibición:* Laboratorio Arte Alameda, Galería Manuel Felguerez del Centro Multimedia, Centro Nacional de las Artes CDMX; Museo Británico-Americano MX. *Material:* Objetos percutidos, Plurifocal Event Controller, fuentes de poder. *Tecnologías creativas:* Juum.

DESCRIPCIÓN

Objetos resonantes es una serie de exploraciones que giran en torno a la generación de relaciones entre objetos percutidos y el entorno. *Objetos resonantes* ha tenido hasta la fecha dos iteraciones que a su vez están relacionadas con otro proyecto que ayudó a su gestación. Para entender la lógica que hay detrás de estas piezas es importante recalcar que *Objetos resonantes* derivó de conceptualizar un proyecto que se llamaría *Ecos para locaciones olvidadas*. Como tal, *Ecos para locaciones olvidadas* nunca se realizó formalmente, pero encarnó en la práctica al proyecto *Objetos resonantes*. Para entender esta derivación revisemos en qué consistía la primera propuesta que fue sometida a un apoyo de producción.

ECOS PARA LOCACIONES OLVIDADAS

Propuesta del proyecto

Ecos para locaciones olvidadas es un proyecto sonoro que busca redescubrir el potencial sonoro de lugares y recintos viejos o abandonados como casas antiguas, bodegas, iglesias y construcciones o arquitecturas en ruinas. Los lugares seleccionados serán intervenidos con una serie de dispositivos inalámbricos capaces de producir sonidos a través de bocinas y por medio de solenoides y motores percutores. El conjunto de estos dispositivos crea una red inalámbrica que, controlada desde una computadora portátil, permite explorar la topología sonora particular de cada lugar y producir experiencias acústicas plurifocales de manera sencilla.

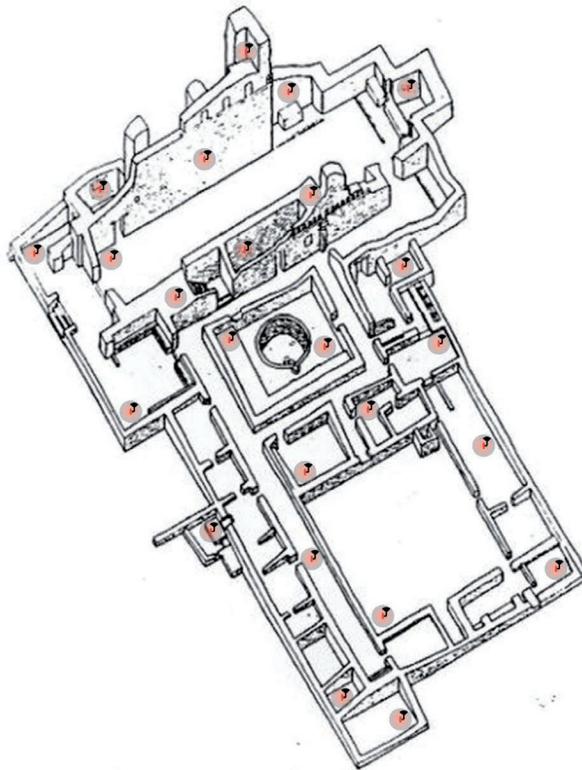


Figura 11.1. | Ruinas de una misión con puntos sonoros distribuidos. Bosquejo original extraído de <http://www.rozylowicz.com/retirement/graphics/missions>.

Para realizar el proyecto será necesario diseñar y producir el equipo electrónico, programar el software de control, seleccionar los lugares y realizar las intervenciones sonoras. Cada etapa del proyecto será documentada y el software y los diagramas realizados serán públicos y accesibles en internet para beneficio de la comunidad de artistas digitales. Mucho del trabajo a realizar se basa en proyectos técnicamente similares desarrollados anteriormente por el artista.

Descripción narrativa

Imaginemos una casa abandonada en el centro de una ciudad o una pequeña iglesia olvidada. Al entrar la recorreremos con la mirada, pero lo que más nos cautiva son los sonidos que proceden de diversos puntos del lugar. Algunos provienen del techo, otros de las esquinas distantes y otros de cuartos alejados. Algunos de los sonidos son “abstractos”, ruidos rugosos, ruidos blancos, ruidos ríspidos y espinados. Otros son, por el contrario, voces, susurros, casi cantos. Aunque los sonidos que escuchamos son variados, su conjunto tiene una lógica acústica y una relación clara con la arquitectura y función del lugar. Nos llama la atención no sólo la textura global sonora, sino sobre todo el juego en el espacio físico que se da entre las diferentes capas acústicas y los puntos sonoros.

Nos detenemos a escuchar y al cabo de unos minutos nos intriga la manera en la que se producen y coordinan los sonidos. Así, pues, nos acercamos a uno de los lugares de donde proviene el sonido y nos encontramos con un pequeño dispositivo electrónico que está alimentado por una batería y que tiene conectados una bocina y un pequeño motor vibrador. Descubrimos que algunos de los sonidos se producen cuando el motor gira y golpea el piso de madera, y que otros sonidos se producen desde la bocina.

Al observar con más detalle vemos que el circuito electrónico claramente no comercial se compone de una microcomputadora, un receptor inalámbrico, un microcontrolador, un amplificador de audio y un puerto analógico en donde se conecta el motor. Al ver el receptor inalámbrico decidimos buscar el transmisor y después de recorrer todo el recinto encontramos una pequeña computadora en cuya pantalla se ejecuta un software no comercial.

Decidimos regresar a nuestra actividad perceptiva y continuamos atendiendo lo que ahora sabemos que es un trabajo sonoro plurifocal creado específicamente para este espacio y parte del proyecto *Ecos para locaciones olvidadas*.

Descripción conceptual

En este proyecto artístico se busca plasmar de forma acústica varias inquietudes y propiedades sonoras de nuestra época, así como proponer una mirada estética a varias de las relaciones entre el espacio físico, la arquitectura y la tecnología. Para ello se propone la realización de un marco sonoro, estético y tecnológico que invite al público a confrontar su realidad acústica y su relación con el entorno físico y sonoro.

Los elementos centrales en la propuesta son:

- Trabajar con el sonido por sus propiedades de abstracción y su plasticidad de reinterpretación.
- Intervenir sonoramente arquitecturas que cuenten con un gran potencial poético y sensorial en las que el sonido pueda recalcar y acentuar significados y simbolismos del lugar.
- Hacer una propuesta artística personal que extienda digitalmente el concepto y la lógica de la música plurifocal.
- Aprovechar las capacidades de las tecnologías digitales para implementar elementos puntuales de la propuesta sonora y que serían difíciles de lograr con otros medios.
- Apropiarse del lenguaje digital por medio de la creación de tecnologías propias y necesarias para el proyecto como parte integral de la propuesta estética.

Como marco de referencia y dejando a un lado a muchos otros autores y trabajos con clara influencia en esta propuesta se hace mención de tres obras diferentes que ofrecen un panorama general del trabajo sonoro en relación con el espacio y la arquitectura. El trabajo del compositor español Llorenç Barber quien maneja el concepto de música plurifocal de grandes dimensiones usando campanarios de iglesias. De manera sencilla y elegante Barber crea una muy valiosa exploración del sonido en comunión con el ritual y la colectividad. Llama también la atención la obra *Playing the Building* de David Byrne la cual, dejando a un

lado el resultado estético, presenta una similitud técnica con *Ecos para locaciones olvidadas*. Otro ejemplo interesante es el trabajo que el joven artista mexicano Yurian Zeron presentó durante la magna exposición para celebrar los 20 años del FONCA en la biblioteca Vasconcelos de la Ciudad de México. En esta obra Zeron percutió a distancia las columnas de la biblioteca mediante solenoides controlados por cable desde una computadora. En este caso existen también claras similitudes conceptuales.

Descripción tecnológica

Para realizar la experiencia sonora es necesario producir un sistema que nos permita la creación y la coordinación de materiales sonoros diseminados en un rango de 100 o 200 metros cuadrados con o sin paredes. Se contempla la creación de 50 puntos sonoros. Cada punto sonoro debe reproducir materiales pregrabados y activar un motor o solenoide que funciona como percutor de alguna superficie u objeto del lugar. Cada nodo debe tener comunicación constante con una unidad central que coordinará la composición sonora.

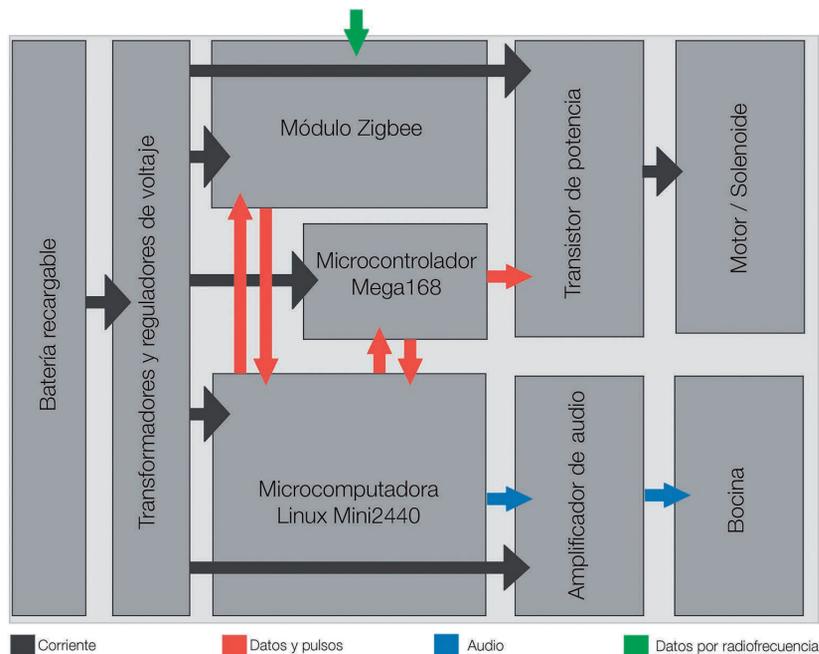


Figura 11.2. | Diagrama de partes del nodo sonoro.

Aunque existen algunas alternativas comerciales ningún sistema se adecua totalmente a las necesidades específicas del proyecto o los costos son muy elevados dada la cantidad de puntos sonoros con los que se desea trabajar. Por tanto, y aprovechando la experiencia en proyectos anteriores, se propone como primer paso el diseño y creación de los circuitos electrónicos requeridos.

Estos circuitos se compondrán de una microcomputadora mini2440 con Linux, un microcontrolador Mega168, un regulador de voltaje, un transistor de potencia para el solenoide/motor, una bocina, un amplificador de audio y un transductor inalámbrico de datos Zigbee. El dispositivo se alimentará de una pila recargable y medirá aproximadamente $15 \times 15 \times 5$ cm (ver figura 11.2).

Se sugiere el uso de la microcomputadora mini2440 debido a su costo –menos de 100 dólares por unidad–, a su tamaño y en especial a sus características, pues funciona con Linux y entre otras opciones tiene convertidores analógicos, audio, puerto para memoria externa, USB y puertos seriales. En proyectos anteriores la he utilizado para controlar motores a distancia, he compilado y utilizado el programa de audio Pure Data y he instalado el sistema Android en ella.

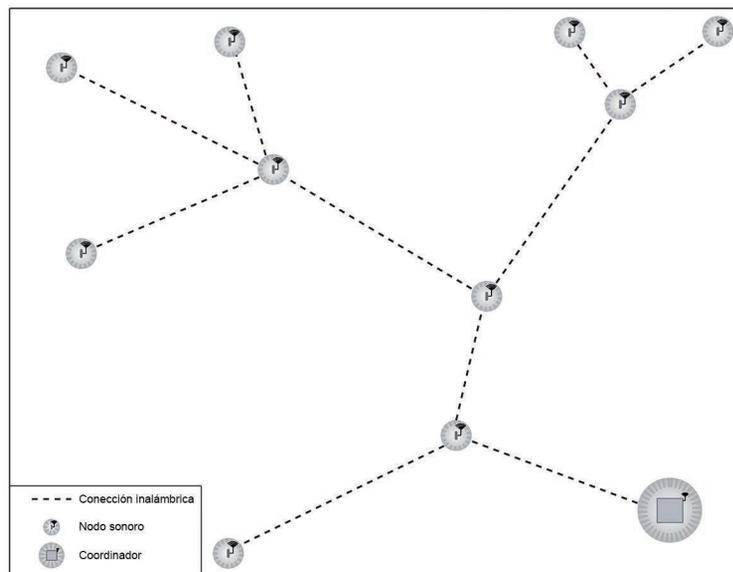


Figura 11.3. | Funcionamiento de la red de datos.

En relación con la transmisión de datos se propone el uso de la tecnología Zigbee, ya que ésta nos permite la creación de redes de datos inalámbricas en el rango de distancia deseada, consume muy poca corriente y permite la implementación de redes mesh con las cuales se podrán cubrir recintos más grandes de arquitecturas complejas en donde la conexión de todos los nodos se tiene que dar en cadena (ver figura 11.3). Para controlar el solenoide o motor se sugiere el uso del microcontrolador Mega168 debido a su costo y sus puertos de comunicación, además de que conozco bien su arquitectura pues lo he utilizado en muchos otros proyectos.

Los circuitos serán diseñados en Eagle, los prototipos serán creados en un Router PCB LPKF usando los programas CircuitCAM y BoardMaster. Una vez que se haya probado el circuito, éste se mandará a producir en la compañía PCBFabExpress. Los componentes electrónicos se comprarán en la tienda en línea Digkey. El montaje de los componentes se hará manualmente uno por uno en el taller. Las figuras 11.4, 11.5 y 11.6 muestran el proceso de creación de un circuito similar para el proyecto *Metáforas para pianos muertos*. La programación del dispositivo consta de varias partes. La microcomputadora se programa en C con las librerías comunes de Linux y el microcontrolador y el módulo Zigbee se programan en C usando AVRStudio.

El software de control y composición será una extensión y adaptación de un proyecto anterior llamado *Juum*. Este programa está creado en Java y Groovy y está concebido como una herramienta para el control temporal y algorítmico de eventos multimedia. Se propone el uso de este programa para dar seguimiento a una propuesta artística en donde el desarrollo de las herramientas técnicas que la generan es parte integral de la obra.

Objetivos

El objetivo principal de este proyecto es crear una serie de intervenciones sonoras que exalten las propiedades, peculiaridades y simbolismos de los lugares escogidos a través del sonido. Se espera que estas experiencias sonoras inviten a los escuchas a experimentar y percibir una realidad poética. Una serie de objetivos secundarios es esperada y concebida como parte del proceso creativo:

- Crear una infraestructura que permita al artista y a otros artistas producir trabajos que requieran los mismos lineamientos técnicos. El sistema está integrado por el software de control y los 50 módulos sonoros inalámbricos.
- Producir documentación, diagramas, esquemas y código de programación detallados que permitan a otros artistas copiar, modificar o extender el software y el hardware que será producido.

PRIMERA VERSIÓN DE OBJETOS RESONANTES

Objetos resonantes 1 es la primera obra de una serie de trabajos que buscan explorar el potencial sonoro de materiales cotidianos resonantes tales como cubetas, botes, latas, cazuelas, etc. En su conjunto estos objetos son convertidos en obras de gran formato empleando tecnologías digitales y mecánicas que permiten percutir los materiales y generar patrones rítmicos desde una computadora de forma inalámbrica.

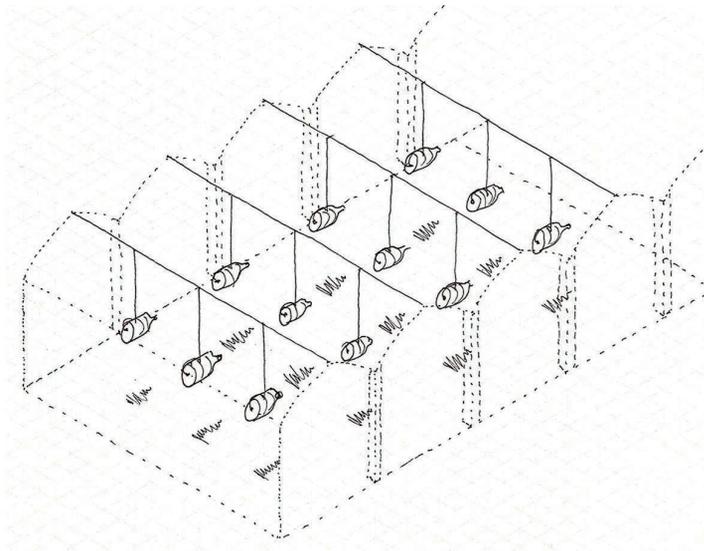


Figura 11.4. | Esquema del montaje de los garrafones.

Para este primer trabajo se utilizan 96 garrafones de agua suspendidos horizontalmente a la altura del oído distribuidos en una matriz geométrica de 16 líneas con seis garrafones

cada una. Cada garrafón tiene montado un solenoide (motor percutor) que activa una baqueta la cual golpea la base del garrafón. En la obra se busca exaltar conceptual y metafóricamente la urbanidad y la plastificación de nuestra sociedad; el garrafón como objeto de uso cotidiano y elemento de nuestro hábitat urbano.



Figura 11.5. | Prototipo de sistema percutor.

La activación de cada garrafón se realiza inalámbricamente desde una computadora a partir de la cual se generan algorítmicamente patrones rítmico-espaciales utilizando el “Sistema Plurifocal” desarrollado por el artista y consta de la tecnología necesaria (módulos electrónicos inalámbricos y software de control) para la creación de obras sonoras en grandes espacios.

Objetos resonantes I fue montada en la nave principal del Laboratorio Arte Alameda cuya sede es una iglesia antigua en el Centro Histórico de la Ciudad de México. La nave es un espacio de 10 metros de ancho por 35 de largo con una altura de 15 metros. Las resonancias,



Figura 11.6. | Montaje de la pieza en el Laboratorio Arte Alameda.

ecos y reverberaciones del lugar fueron ideales para esta obra sonora, pues los desplazamientos sonoros y los patrones rítmicos fueron exaltados y complementados con las propiedades sonoras del recinto.



Figura 11.7. | Objetos resonantes en el Laboratorio Arte Alameda.

La obra fue realizada mediante el Programa de Apoyo a la Producción e Investigación en Arte y Medios 2010 del Centro Multimedia del Centro Nacional de las Artes. La obra se produjo para el programa *Arte-Shock* de TV UNAM. La obra recibió el patrocinio de ePura (Pepsico) empresa que proporcionó los garrafones.



Figura 11.8. | Vista superior de *Objetos resonantes* en el Laboratorio Arte Alameda.

SEGUNDA VERSIÓN DE OBJETOS RESONANTES

La segunda versión de la obra fue producida para el festival *Transitio_MX*. En esta segunda versión se utilizaron, como fuentes sonoras, cubetas metálicas de pintura. La generación sonora era más caótica. El sistema digital fue actualizado y permitió el control de la intensidad sonora, lo que dio pie a una paleta mucho más rica de sonoridades y recursos tímbricos. ■■■



Figura 11.9. | Montaje de la pieza en el Museo Británico Americano.



Figura 11.10. | *Objetos resonantes* en el Museo Británico Americano.



Axial

Hoja de sala

Fecha de producción: 2014. *Lugares de exhibición:* Jardín Botánico de la Universidad de Washington. *Material:* Contenedor de barco, submarino osceanográfico, hidrófonos, cómputo en la nube. *Tecnologías creativas:* SuperCollider, Plurifocal Events Controller, Juum.

QUÉ ES AXIAL

A*xial* es una instalación que recurre al sonido para representar de forma abstracta el poder simbólico del océano y del mar profundo. Su nombre proviene de la Cordillera Volcánica Axial ubicada en la cresta subacuática Juan de Fuca en donde se realizaron las grabaciones sonoras y se obtuvieron los datos oceanográficos para su creación. Estos últimos fueron obtenidos gracias al apoyo y a la ayuda de la iniciativa Ocean Observatory que, a su vez, se benefició del apoyo de la National Science Foundation. El material obtenido durante la expedición oceanográfica fue interpretado y convertido en una instalación sonora posterior, utilizando grabaciones, entre las que destaca un conjunto de registros de sonido de ventilas hidrotermales situadas en el fondo marino. La instalación sonora que funcionó como receptáculo final consistió en un contenedor de carga empleado como resonador sonoro que, además de su función resonadora, operaba como delimitador físico el cual –

conceptualmente— representó el vacío y marcó esa frontera (línea divisoria) entre el mar y el resto del medio ambiente.

Además de lo descrito anteriormente en relación a su parte física y la generación del sonido, *Axial* es una exploración auditiva donde dos capas acústicas distintas se unen y dialogan para generar la experiencia acústica, pero ¿cómo opera? Por un lado, un conjunto de 80 solenoides con baquetas uniformemente distribuidos golpean las paredes del contenedor de carga, produciendo patrones rítmicos y, al mismo tiempo, una segunda capa de material digital es generada mediante seis altavoces que cuelgan de las paredes del contenedor. Para que la experiencia resulte óptima el contenedor se coloca en un campo abierto con el fin de maximizar la sensación de vacío, distancia y territorio abierto y, después de una búsqueda consciente, se decidió ubicarlo en el Centro de Horticultura Urbana de la Universidad de Washington que, como escenario para la instalación, reunía las condiciones de un paisaje despejado, abierto y en contacto con la naturaleza.

La creación de este trabajo requirió una exploración artística que duró más de dos años y, por tanto, *Axial* está en deuda con un amplio conjunto de influencias entre las que se incluyen la obra de Juan Pampin, Richard Karpen y James Coupe. Esta obra transita desde distintos paradigmas pero, sobre todo, a través de la obra de Trimpin y basado en las ideas sonoras de Lloren Barber, que explora el sonido en el espacio público (véase la cronología de conciertos y ciudades). Además de la exploración estética de la obra, para su realización tuve que crear un software y un conjunto de dispositivos electrónicos fueron elaborados y desarrollados específicamente para producir este trabajo.

Axial es una pieza compleja de múltiples capas y está directamente influenciada por muchas obras y propuestas anteriores de numerosos artistas y creadores, pero responde a mi interpretación personal ante preocupaciones artísticas muy concretas que incluyen el significado del espacio público, la estética de la sonificación de datos, el potencial y las limitaciones de la música mecánica, el significado de la distribución sonora *plurifocal* y, sobre todo, a las intersecciones entre arte, ciencia y tecnología. Este capítulo describe tales preocupaciones señalando la referencias artísticas e influencias directas a la obra.

Como se dijo anteriormente, *Axial* es una obra creada durante un largo periodo y, por ello, pasaron varios años entre la idea seminal y el despliegue de su versión final. Durante el

periodo de su concepción, gestación y realización probé varias propuestas que fueron sometidas a discusión y análisis. *Axial* no floreció de la nada, mi preocupación por trabajar con el océano y con el mar profundo provenía de mucho tiempo atrás. Por un lado tuve la oportunidad de realizar una expedición oceanográfica inicial –durante mis estudios de bachillerato, en 1993–, y, por otro, la expedición Enlighten en el año 2010 que reforzó mis intereses, demostrando que la idea germinal seguía allí contenida. Es importante mencionar que entre estos dos acontecimientos produje varios proyectos relacionados con el agua, los viajes y el sonido, y esos procesos creativos son los que describo a continuación.

EXPLORACIÓN Y BOCETOS

La creación artística y probablemente todos los campos de la creatividad humana rara vez son un camino de exploración lineal y directo; por lo general, el descubrimiento humano –y especialmente el descubrimiento artístico– tiende a ser un camino sinuoso con cambios de dirección, actualización de ideas e incorporación de descubrimientos que en cada uno de los pasos aportan al desarrollo de una idea. Aunque la lógica de *Axial* estaba clara desde el principio, el



Figura 12.1. | Expedición como parte del proyecto *Jason*, en 1993.

proceso de realización experimentó numerosos cambios durante el desarrollo. Honestamente, durante el proceso creativo inicial no estaba claro del todo cuál sería el mejor medio para mostrar el concepto de la obra y, por esta razón, se exploraron distintas alternativas durante varios meses. Estas elecciones se basaron en las habilidades que adquirí durante los años previos y en la exploración de las posibilidades más cercanas al ejercicio estético.

Expedición Jason

Mientras realizaba mis estudios de bachillerato en el Colegio de Ciencias y Humanidades de la Universidad Nacional Autónoma de México tuve la oportunidad de participar en una expedición oceanográfica. El proyecto *Jason* (<http://www.jason.org/>) es un programa de ciencias de la escuela media superior de los Estados Unidos iniciado en 1989 por el Dr. Robert Ballard con el objetivo de alentar a los estudiantes a explorar los campos de la ciencia y la tecnología. Cada año el proyecto *Jason* desarrolla una expedición organizada en torno a un interés científico particular; en 1993 la expedición fue al Golfo de California para estudiar las ventilas hidrotermales en la Bahía de Concepción y fui beneficiado con la oportunidad de participar.

Los estudiantes pasamos una semana desarrollando un proyecto de investigación con el apoyo del equipo científico, al mismo tiempo que explorábamos y aprendíamos a manejar el equipo oceanográfico. Durante ese lapso tuve la oportunidad de ayudar al equipo de ROV (Remote Operated Vehicle) y participar en las observaciones de las ventilas hidrotermales.

Casualmente, durante este tiempo mis intereses en la composición musical se profundizaban y las imágenes, los sonidos y la experiencia ante la exploración del fondo marino las abordé no sólo desde un punto de vista científico, sino como un detonador de imágenes artísticas que impulsaban mi experiencia estética. Me tardé casi 20 años para materializar esas imágenes iniciales que se gestan en el instante en el que una obra de arte es concebida. Hoy en día para mí resulta muy evidente lo definitorio que fue la experiencia del proyecto *Jason* en aquellos momentos y reconozco que fue una poderosa influencia en un nivel subconsciente para el desarrollo posterior del proyecto *Axial*; y gracias a las reminiscencias de aquella exploración inicial la obra cuenta con observables y rastreables concretos que toman forma como experiencia estética.

Recorrido marítimo y el barco de madera

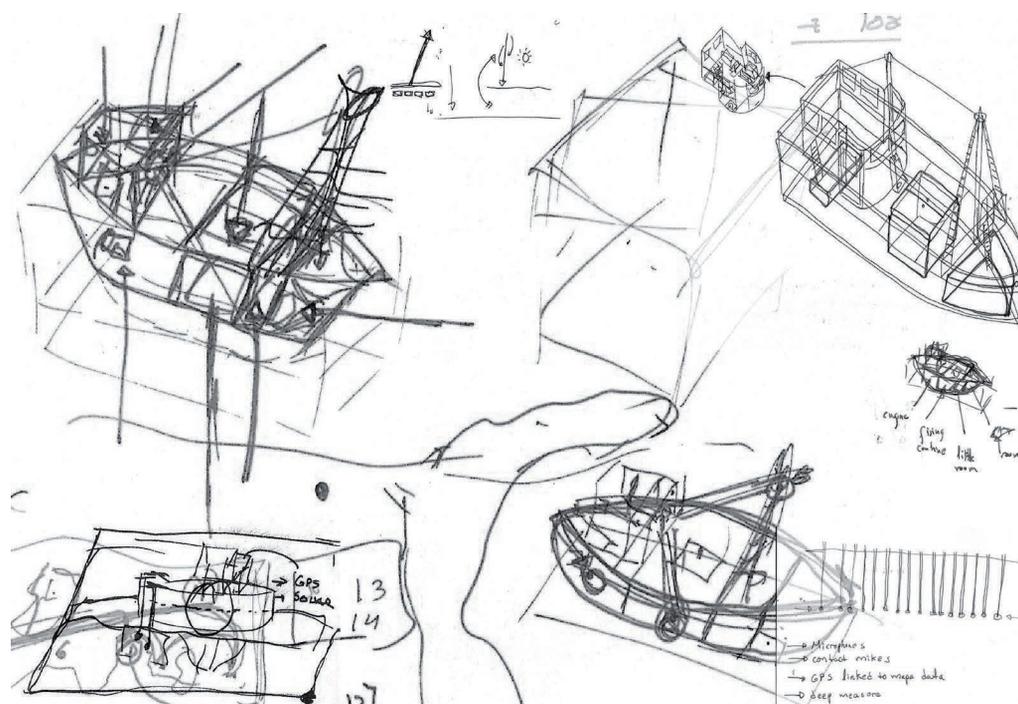


Figura 12.2. | Primer boceto del recorrido sonoro marítimo.

La idea inicial de *Axial* comenzó durante un curso de arte mecatrónico –impartido por James Coupe en el Centro de Arte Digital y Medios Experimentales de la Universidad de Washington–, porque durante este curso yo mismo intenté generar una experiencia sonora en un barco de madera visitando el puerto pesquero de Seattle, preguntando cuestiones específicas a los propietarios y a los conductores de los barcos. Durante este instante de curiosidad me puse en contacto con un capitán de barco nativo americano que estaba dispuesto a alquilarme un viejo barco de madera para hacer una travesía sonora que funcionara como experiencia auditiva; de hecho, el mismo capitán me invitó a ver el barco en Bainbridge Island y, de su voz, obtuve algunas narraciones (historias y leyendas de marineros nativos americanos de la región) que tenían el potencial de convertirse en líneas narrativas del proyecto. Las negociaciones con el capitán de barco se interrumpieron en algún momento, pero la idea de crear

un viaje sonoro que funcionara como salida artística se hizo más fuerte en mi interior¹ después de aquella búsqueda. En la figura 12.3 muestro la propuesta inicial que llevaba por título *La voz del barco* y que fue la que se presentó durante el curso de arte mecatrónico:

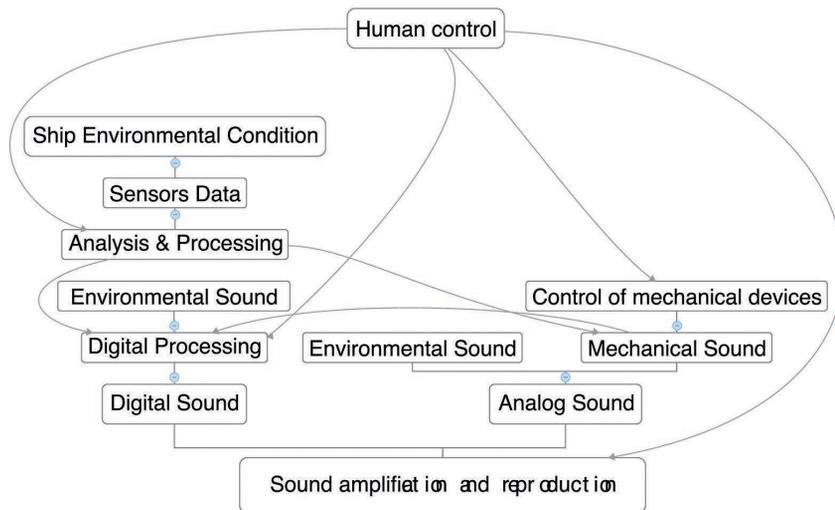


Figura 12.3. | Esquema del recorrido como fue planeado para el curso de arte mecatrónico.



Figura 12.4. | La halibut schooner Silver Wave.

¹ La pieza *Rheinklänge* de Christina Kubisch sigue la misma lógica de acción.

La motivación detrás de *La voz del barco* se basaba en generar una experiencia sonora original donde el objeto que *cantara* fuera un barco. El plan consistió en desarrollar un software personalizado, así como un hardware especialmente diseñado para el proyecto, que produjeran la experiencia sonora en tiempo real. El diagrama de flujo de la obra es el siguiente: una serie de micrófonos de contacto se colocarían en diferentes estructuras de resonancia del barco y, adicionalmente, se pondría un solenoide cerca de cada micrófono que permitiera activar la generación del sonido utilizando el software específicamente diseñado para el proyecto. Para que esto fuera viable el *ejecutante* controlaría la actividad de los solenoides y las transformaciones de la señal proveniente de los micrófonos en tiempo real. En este proceso la *composición* se transmitiría en dos altavoces situados en la cubierta del buque y un equipo de artistas sonoros estaría invitado a producir material, realizando una interacción sonora durante un corto recorrido en alta mar. En esta experiencia los pasajeros son, en simultáneo, los espectadores-receptores del evento sonoro y colaboradores de la experiencia.

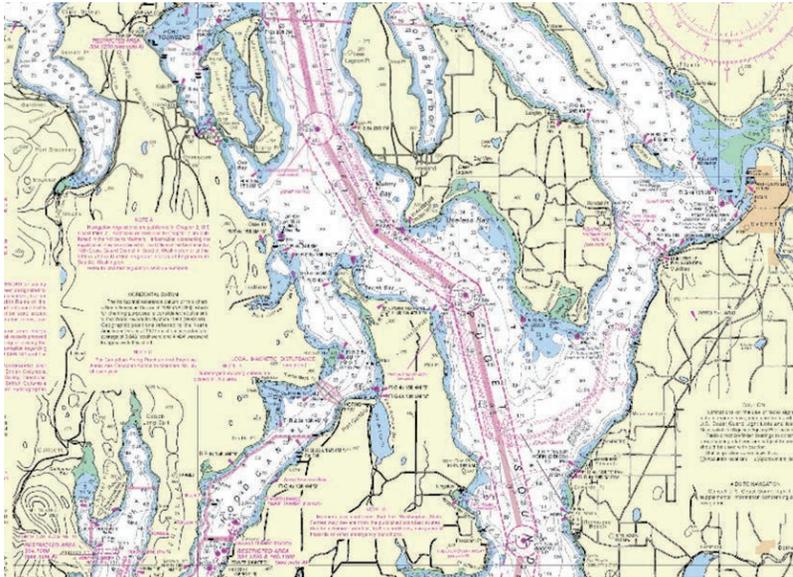


Figura 12.5. | Carta náutica de la región de Puget Sound.

Exhalation-Water

El segundo trabajo que antecede a *Axial* fue un proyecto que funcionó como espacio de experimentación titulado *Exhalation-Water* y que ya describimos en el capítulo 7. Dicho proyecto fue creado bajo una línea curatorial específica para la cual elegí el diseño de una experiencia sonora que genera un diálogo entre el espacio, el sonido, el ambiente y el movimiento (no sólo de los participantes), sino también de los objetos físicos con propiedades sonoras a lo largo y ancho del espacio donde ocurre el recorrido sonoro.

Exhalation-Water está concebido con varios hilos lógicos creativos que continuaron desarrollándose a lo largo de la conceptualización, el diseño y la realización de *Axial*. A pesar de que la versión final de *Axial* no es un viaje sonoro en el sentido de que no existe un desplazamiento físico de la audiencia durante la presentación, las ideas de la travesía y del movimiento continuaron siendo una parte importante del concepto artístico para la presentación de la obra. En otras palabras, el *viaje* como una idea metafórica y conceptual comenzó ante el proyecto *Exhalation-Water* y permanece presente como *travesía* (en un formato alterno) que se convirtió en *Axial*. En este sentido, *Axial* es una reminiscencia y una transfiguración de los viajes que realicé durante el proceso de investigación creativa a bordo del buque *Thompson*, que explicaré más adelante.

Primera expedición, imágenes

Durante el verano del año 2009, participé en la expedición oceanográfica Lumens como parte de la iniciativa Ocean Observatory. Desafortunadamente, la expedición que partió de la Universidad de Washington sólo alcanzó a llegar a las esclusas de Ballard antes de que el barco *Thompson* experimentara problemas mecánicos serios y el crucero fuera cancelado. A pesar de que se canceló la expedición, los investigadores pasamos un par de días a bordo del barco ante una serie de conferencias y charlas que tuvieron lugar ahí mismo. Durante este periodo tuve oportunidad de realizar un conjunto de grabaciones sonoras y una intensa documentación fotográfica que fue una referencia visual detonadora muy importante del proceso. El diálogo entre el sonido en el barco, así como los espacios arquitectónicos del mismo y las experiencias sensoriales con el océano comenzaron durante este viaje, y el sim-

ple hecho de caminar, escuchar y entender las imágenes de la región de Pudget Sound me ayudó para contar con una idea más clara sobre cómo abordar la encarnación de la obra artística en relación con el mar. Gracias a esta aproximación pude diseñar una serie de propuestas para llevar a cabo durante la expedición programada para el año siguiente.



Figura 12.6. | Fotografía tomada durante la expedición *Lumens* (A).



Figura 12.7. | Fotografía tomada durante la expedición *Lumens* (B).



Figura 12.8. | Fotografía tomada durante la expedición Lumens (c).

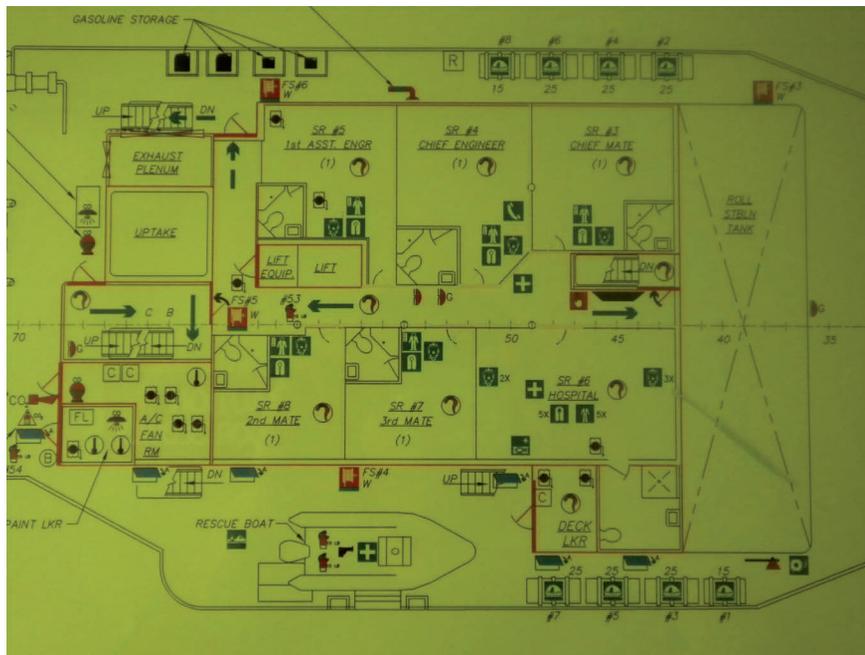


Figura 12.9. | Fotografía tomada durante la expedición Lumens (d).

PROPUESTAS INICIALES

Propuesta 1: Laboratorio para performance en alta mar

La primera idea para la implementación del concepto fue crear una cabina *expresiva* de comunicación en la cubierta de un barco oceanográfico adaptado dentro de un contenedor de carga. Este espacio sería el lugar físico para crear y transmitir al mundo exterior contenido sonoro; allí mismo se colocarían diferentes tipos de objetos –generadores capaces de producir sonidos estridentes (palos, tubos, martillos y placas metálicas, entre otros)–. Para lograr esta capa sonora en forma de improvisaciones de audio trabajé a lo largo de la expedición oceanográfica como agente traductor de las impresiones e interacciones entre la actividad humana del barco y la audiencia en tierra.

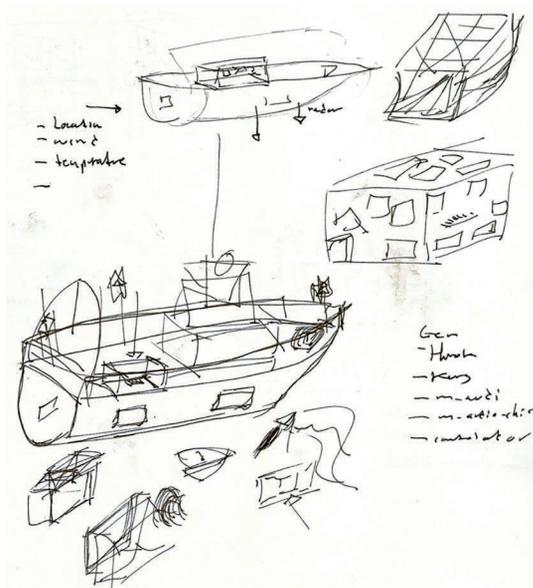


Figura 12.10. | Boceto de la propuesta 1. Laboratorio para *performance* en alta mar.

Inspirado en la obra *Cremaster 3* de Matthew Barney (<http://www.cremaster.net>), la idea de la propuesta consistía en crear una narración de imágenes con un alto nivel de abstracción que, a su vez, fueran un recorrido (trayectoria de experiencias físicas moldeadas

das por el cuerpo humano) y al mismo tiempo que tuvieran en sí mismas constantes referencias y alusiones al mar. Los *performances* no serían predefinidos con anterioridad y surgirían de las acciones y de los eventos que ocurrieran en el barco en estrecha relación con las situaciones ambientales.

Para que esto sucediera fue importante actuar como observador testimonial de la vida a bordo del buque y responder con improvisaciones sonoras utilizando fuentes de sonido (análogas y digitales) expandidas que tuvieran una comunicación digital en tiempo real. La improvisación/*performance* sería transmitida a través de internet de banda ancha y, gracias a ello, la gente sería capaz de ver la actividad y los productos audibles resultantes a través de la web.² Aunque algunos elementos de esta propuesta fueron adoptados en la versión final, hubo varios obstáculos para la realización de esta versión, ya que había preocupaciones sobre la factibilidad del trabajo, sobre todo, en relación a la estabilidad de las conexiones de internet a bordo y sobre la logística de las posibilidades de mover un contenedor cargado con instrumentos musicales dentro del barco de investigación científica. Al inicio no fue evidente que la propuesta considerara la intención de manejar el proceso tecnológico en tiempo real, y, por tanto, el elemento de improvisación parecía ser más un signo de desorganización que un mecanismo sólido para la producción de sonido; además, había preocupaciones acerca de los periodos vacíos de datos (aquellos huecos de poca o nula actividad durante el día cuando no hubiera eventos ni improvisaciones por representar). Con este antecedente se diseñó una segunda encarnación de la propuesta que describiré a continuación.

Propuesta 2: Experiencia telemática multicanal

Adaptar la propuesta inicial a otro formato fue un proceso muy complejo, principalmente porque algunas de las ideas de la primera versión se fijaron en mi imaginación y me costaba mucho trabajo soltar la idea germinal tan fácilmente. Me fue muy difícil abandonar algunas de las opciones que originalmente parecían buenas selecciones en la producción de la pieza y, por esta razón, la segunda propuesta intentó mantener varias de las ideas centrales de la primera.

² No se puede dejar de pensar en la propuesta de Stelarc.

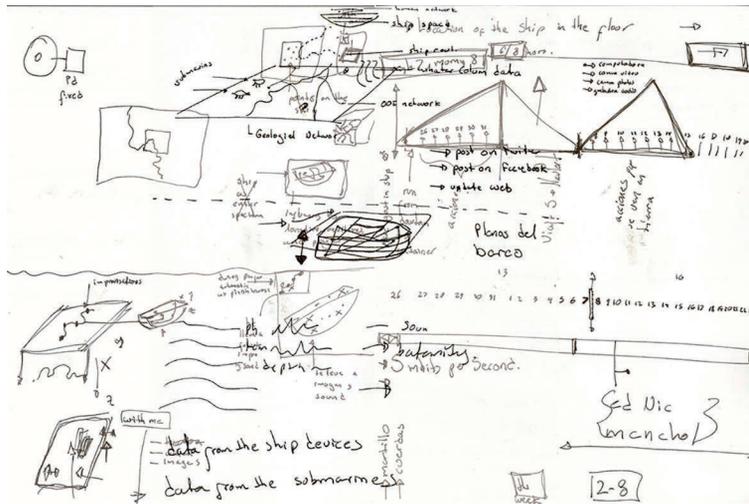


Figura 12.11. | Diagramas de la propuesta 2. Experiencia telemática multicanal.

Esta segunda propuesta movía el centro de comunicación y de las actividades performáticas del barco a tierra, por lo cual propuse recibir los datos científicos producidos por el buque oceanográfico y utilizarlos para modular los parámetros de las acciones cotidianas que debían suceder en el contenedor de carga en tierra y que éste a su vez fuera el receptáculo de la obra. El contenedor de carga se colocaría en un campo abierto y se utilizaría como centro receptor de información, captando datos del buque a través de una conexión a internet; aunque en cierto momento consideré el hecho de habitar el contenedor, vivir durante una semana dentro de él sin poder salir, con la intención de que se creara en mí un estado sensorial, anímico y fisiológico particular para convertir en territorio de sentido mi contenedor.

Fue bajo este contexto e intención que decidí ofrecer una improvisación sonora que ocurriera al atardecer, utilizando el contenedor como escenario y que recibiera una influencia directa por el acontecer y por las novedades y variaciones en la información generada en el barco.³

Aunque para este momento se sabía que el análisis y la transformación podrían ser realizados mayormente en los servidores de Amazon Cloud, el aspecto técnico de esta versión

³ Esta propuesta guarda cierta semejanza con el trabajo de Coco Fusco y Guillermo Gómez Peña (dos indígenas no descubiertos) (http://en.wikipedia.org/wiki/Coco_Fusco).

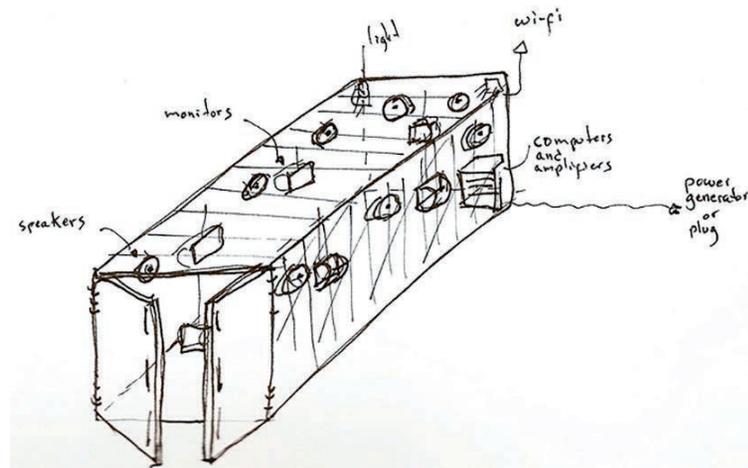


Figura 12.12. | Boceto de la propuesta 2. Experiencia telemática multicanal.

proponía el uso de microcomputadoras, específicamente la Mini2440 (<http://www.friendlyarm.net/>) porque ofrecía una solución alternativa y económica para el procesamiento de datos. Estas microcomputadoras conformarían un sistema embebido dentro del contenedor evitando el uso de parafernalia digital grande y obvia dentro del contenedor.

Esta versión la analicé y trabajé con colegas y con artistas digitales, sobre todo, porque tenía la sensación de que en esta propuesta descansaban demasiadas ideas, demasiados medios y demasiadas situaciones indefinidas que no le daban claridad, así que para este punto decidí suspender el proyecto durante algunos meses con el fin de decantar algunas de las ideas y filtrar los elementos que no fueran necesarios y que, al parecer, estaban diluyendo la idea central. Fue hasta después de dicha pausa que, con la influencia de las obras creadas durante esos meses, la versión final fue concebida y ejecutada óptimamente.

PREOCUPACIONES ARTÍSTICAS EN AXIAL

Después de las etapas iniciales de experimentación (ensayo y error), así como del proceso de maduración conceptual, la pieza alcanzó su manifestación final en el otoño de 2011. La experiencia *performática* se dejó de lado y el *gesto sonoro agresivo* surgió de manera natural

porque apareció como traducción sensorial de la fuerza marítima, del poder geológico y de la magnificencia circundante de los océanos. Al mismo tiempo, la obra adquirió una sencillez de exposición con el objetivo de mantener un diálogo orgánico entre la obra y su entorno natural.

Como obra compleja, la pieza tiene varias capas de interpretación y varias relaciones entre los elementos constitutivos internos lo cual arroja varias lecturas interpretativas, según la perspectiva estética del espectador, y desdobra muchas intersecciones entre la lógica interna de la pieza y la realidad física circundante donde se materializa y toma forma.

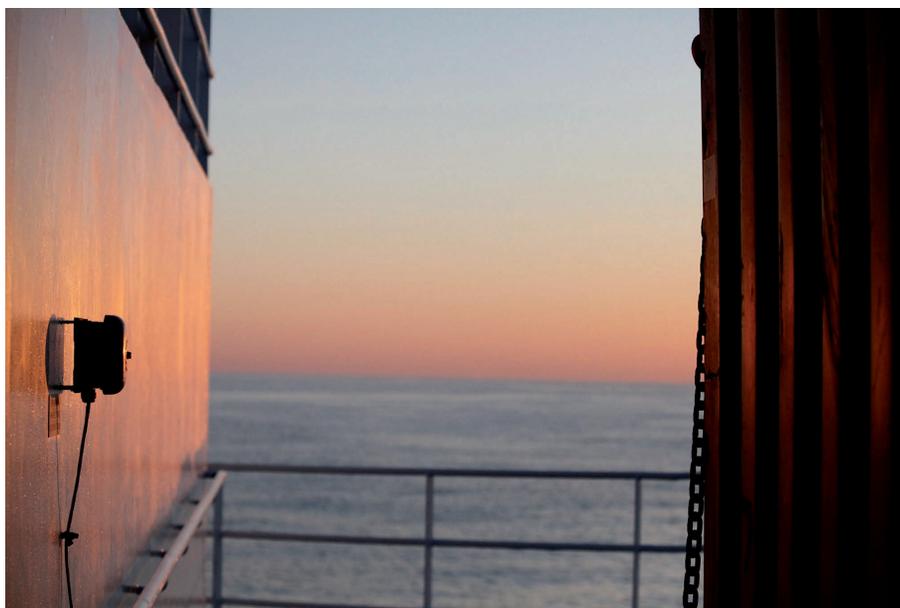


Figura 12.13. | Contenedor en la nave. Esta imagen representa los límites del espacio y la ubicación.

Como ya se ha mencionado y como se desprende de las exploraciones y bocetos que condujeron a esta versión final, el trabajo explora soluciones, ofrece respuestas artísticas y genera conocimiento artístico con las siguientes preocupaciones en mente.

Exploración del sonido Este trabajo utiliza el sonido como medio para representar y manifestar elementos ocultos y metáforas de la realidad visual (tanto oculta como visible). De

esta manera, la grabación de las ventilas hidrotermales y el movimiento del brazo robótico son parte de esta curiosidad nunca satisfecha del arte sonoro.

Dimensión, proporción y frontera La pieza explora las nociones de dimensión, proporción y frontera para representar la relación perceptual del océano entre la humanidad y el resto del mundo.

Espacio y vacío La pieza juega, a nivel metafórico, con los contenidos ocultos y existentes en el vacío; recreando todo aquello que existe abajo de la superficie (específicamente el mar). En esta línea la pieza representa un gesto de curiosidad y de descubrimiento en simultáneo.

Gestos mínimos La pieza indaga sobre gestos reducidos y minimalistas en las estructuras composicionales (los patrones rítmicos, las formas y el formato de visualización) como mecanismo para evidenciar la simplicidad y la evolución a lo largo del plazo de observación de algunos procesos naturales, en particular los geológicos.

Arte, ciencia y tecnología Esta obra explora la intersección del arte, la ciencia y tecnología como un campo híbrido para buscar y desenterrar las relaciones expresivas entre lo humano y lo ambiental.

PAISAJE SONORO SUBMARINO

Preparación de la grabación

En la expedición oceanográfica de 2011 se integró al equipo de investigadores el geólogo marino Timothy Crone. El doctor Crone había grabado señales acústicas en las chimeneas de varias ventilas hidrotermales como una forma de estudiar el flujo a través de las estructuras de sulfuro, proponiendo que los cambios en los componentes de banda ancha fueran indicativos de las variaciones de la velocidad de descarga relacionadas con la mecánica de la carga de la marea (Crone *et. al.*, 2006).

Aunque su preocupación no era musical ni artística Crone había grabado y analizado señales acústicas en las chimeneas como una forma de medir los patrones de las mareas a fin de sugerir implicaciones biológicas. Para realizar estas grabaciones el Dr. Crone y su

equipo diseñaron y fabricaron un sistema de grabación acústica capaz de producir grabaciones de larga duración y de grabar frecuencias por debajo de 100 Hz con muy buena calidad. Entre 2004 y 2005 el equipo de Crone produjo dos versiones de la grabadora y capturó varias grabaciones de ventilas hidrotermales. La tabla 12.1 (Crone *et. al.*, 2006) muestra las especificaciones de las dos grabadoras diseñadas por el equipo de Crone.

Tabla 12.1

	<i>1st Generation (2004)</i>	<i>2nd Generation (2005)</i>
Microcontroller	Persistor CF2	Persistor CF2
Hydrophone Element	Bentos AQ-2000	Bentos AQ-2000
Sample Rate	1000 Hz	1920 Hz
Number of Channels	3	2
A/D Converter	12-bit (R212)	16-bit (OES AD24)
Anti-Aliasing	3-pole Bessel (-3 dB at 500 Hz)	3-pole Butterworth (-3 dB at 500 Hz)
Programmable Gain	No	Yes
Overall Gain	-161 dB re 1 V/ μ Pa	-181– -145 dB re 1 V/ μ Pa
Dynamic Range	\approx 72 dB re 1 μ Pa	\approx 127 dB re 1 μ Pa
Storage Medium	4 GB Compact Flash	4 GB Compact Flash
Recording Capacity	\approx 10 days	\approx 6 days

Después de aproximarme a la investigación de Crone me fasciné con la posibilidad de producir grabaciones de campo sonoro en el fondo marino, especialmente en las ventilas y en las áreas circundantes. Skip Denny y el equipo de ingeniería de OOI RSN me proporcionaron hidrófonos para realizar las grabaciones iniciales durante un periodo de pruebas en el lago Washington y en Puget Sound. Algunos hidrófonos funcionaron mejor que otros, pero todos registraron con total calidad motores y propelas de barcos, sobre todo, porque las fuentes del sonido estaban muy cerca. Gracias a esta experiencia me familiaricé con los dispositivos, con el equipo y con las propiedades acústicas de la grabación de sonido en agua. La grabación por debajo del agua produce diferentes calidades de material sonoro, especialmente respecto al eco, a las resonancias, a la reverberación y a las frecuencias bajas.

Tim Crone y su equipo llegaron a Seattle un día antes de la partida de la expedición Enlighten 2010. El Dr. Crone había cambiado su técnica de registro y ahora utilizaba grabacio-



Figura 12.14. | Preparación de la grabadora en la cámara presurizada.

nes de video, sobre todo, visión por computadora y procesamiento de imagen para hacer sus referencias en cuanto a los flujos de presión en las chimeneas de las ventilas. Su trabajo sonoro durante el viaje había sido la prueba y ahora recurría al uso del video y del análisis de la visión por ordenador para calcular e inferir la velocidad de flujo en las ventilas y las chimeneas; no obstante, a petición mía, Crone llevó consigo una de las grabadoras de audio y pude trabajar con el equipo necesario para realizar las grabaciones de sonido.

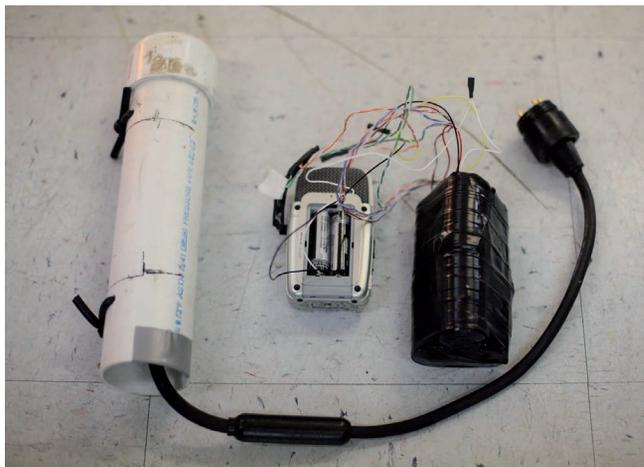


Figura 12.15. | Grabadora Zoom e hidrófono.

Durante los primeros días a bordo de la nave, Crone me explicó los procedimientos y las técnicas para configurar la grabación incluyendo el montaje, la colocación en el ROV, la recuperación y el tratamiento de los datos digitales con Matlab. Además de ayudarlo con sus experimentos de video pude instalar la grabadora conocida como VentPhone 2.0. y con ella grabar hasta 48 horas de audio ininterrumpidas. Por desgracia, el VentPhone tiene una frecuencia de muestreo de 1 920 Hz y aunque esta tasa de muestreo era buena para los propósitos originales de la grabadora en donde sólo se buscaba medir el flujo de presión, la calidad no era ideal para señales en el rango audible y, debido a esta dificultad, decidimos utilizar una segunda grabadora de audio comercial que acompañara conjuntamente al VentPhone.



Figura 12.16. | VentPhone e hidrófono.

El Zoom H2 es un grabador portátil de audio comercial y graba en una tarjeta de memoria externa. Es un dispositivo que puede grabar archivos MP3 comprimidos de tamaño considerable, dependiendo del tamaño de la tarjeta de memoria. Dado que la grabadora no está diseñada para grabaciones de duración extrema, las baterías no duran la cantidad de horas requerida; por tanto, diseñamos un conjunto de baterías externas que nos permitiera una grabación de 48 horas continuas de audio. Con esta solución ambas grabadoras se colocaron dentro del contenedor presurizado y este contenedor se unió al tripié.



Figura 12.17. | Colocación de grabadora de audio y video en tripié.

El 17 de agosto, cuando el buque oceanográfico *Thomson* se localizaba justo sobre la ventila hidrotermal *Medusa*, el trípode fue lanzado en caída libre con el equilibrio adecuado de pesos y boyas para calibrar la velocidad de caída. Una vez que el trípode estuvo en el fondo marino, el Jason Remote Operated Vehicle (ROV) lo movió a un lugar preciso en frente de la ventila y con sus brazos robóticos Jason sacó la tapa de la cámara de video y colocó el hidrófono en frente de la ventila hidrotermal a un lado de la chimenea.

Todas las maniobras del Jason son controladas remotamente desde la cubierta del barco por los conductores del ROV y utilizan un conjunto de cámaras y audio para asistencia visual en tiempo real. Pude acompañar toda la operación de colocación en las pantallas dentro de la sala de control y cuando vi el brazo del ROV moviendo el hidrófono para colocarlo en el ventila pude imaginar el gesto artístico en toda su completud. Para colocar un micrófono en esa remota ubicación de la Tierra y realizar una grabación de sonido de esas propiedades se requería una cantidad de planificación nada trivial y una logística, infraestructura y los recursos propios de la gestación. Tal acción tenía el potencial de convertirse en una declaración y en el gesto artístico primigenio. La imagen de sostener el micrófono en el fondo marino a donde muy pocas personas han tenido



Figura 12.18. | Brazo de Jason colocando el hidrófono en la ventila hidrotermal.

acceso era expresivamente tan intensa que tuve que preguntarle al científico si sería posible disponer de algún tiempo para guiar al conductor del roV en el momento de la recuperación. Afortunadamente, mi propuesta fue aceptada y el gesto performático sería grabado tres días después.



Figura 12.19. | Control de Jason desde la cabina en cubierta.

El gesto de grabación

El 20 de agosto, durante la inmersión J2-518 de Jason, antes de la recuperación del trípode, había un espacio de tiempo para el experimento sonoro y mientras todas las cámaras de video estaban grabando desde diferentes ángulos, allí, en el momento justo enfrente de la ventila hidrotermal, le pedí al operador del ROV que tomara el micrófono y que lentamente alcanzara la punta de la ventila de donde emergía el flujo de agua caliente, luego le pedí que moviera el micrófono hacia adelante, hacia atrás y en círculos, muy lentamente, sobre todo, alrededor de la chimenea de la ventila, imitando el tipo de gestos que un artista sonoro emplea al hacer grabaciones de campo en tierra firme.



Figura 12.20. | Cámaras de Jason registrando el gesto de grabación.

Con esta acción me proponía hacer evidentes las propiedades *performativas* del evento de la grabación y, además, generar un registro narrativo de la situación. Como ya hemos mencionado, la grabación de sonido en el fondo marino no es trivial y el hecho de colocar una grabadora a 1 543 metros por debajo de la superficie para satisfacer una curiosidad puramente sonora fue un desafío, un esfuerzo colaborativo y representó una comunión entre el sonido, el arte y la ciencia. La acción, que más tarde fue editada como un breve video, representa la curiosidad sonora jamás satisfecha de los artistas de sonido que transformamos todo tipo de situaciones, eventos y acciones en expresiones artísticas donde el sonido juega un papel central y protagonista. El video de dos minutos se convirtió en un elemento visual que funciona como un referente documental sobre el trabajo, así como un manifiesto sobre el arte sonoro.

Una vez que la acción performativa terminó el trípode fue recuperado, atando las boyas que permitieron que el equipo regresara a la superficie. El trípode se recuperó con la grúa del buque y allí venían contenidos los materiales para ser estudiados y analizados posteriormente en el barco.

El material grabado

Las grabaciones del fondo marino fueron recuperadas para su procesamiento, edición y manipulación en el laboratorio del barco. La duración total de la grabación fue de 30 horas y 40 minutos. La señal muestra datos claros del periodo de lanzamiento, del instante cuando se desplegó el trípode y también del momento de colocación de la grabadora en el sitio preciso. Después de este momento la señal se volvió débil y escasa, pero gracias al procesamiento digital que incluyó compresión, ecualización, normalización y filtrado la señal presentaba algunos de los sonidos de la ventila hidrotermal. La grabadora VentPhone ofreció una señal limpia sin muchos armónicos debido a la resolución del muestreo, pero el material grabado fue extremadamente valioso debido a la naturaleza del evento.

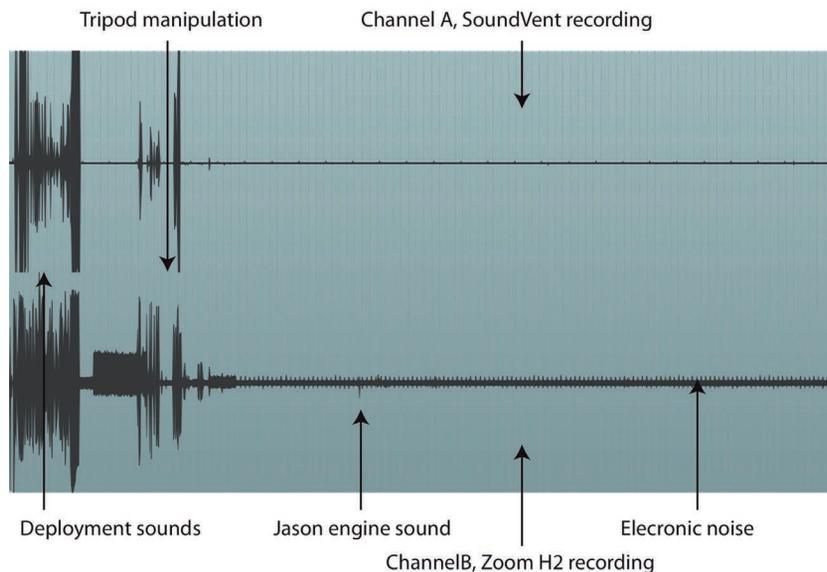


Figura 12.21. | Forma de onda de las primeras 30 horas de grabación submarina. El canal A muestra la grabación de la VentPhone y la grabación B muestra el registro de la grabadora H2.

Desafortunadamente, el registrador H2 presentó una interferencia intermitente constante de ruido generada por los circuitos electrónicos del VentPhone. Tener tal interferencia en el registrador H2 hizo difícil extraer muestras de sonido de larga duración, sin embargo, contamos con ventanas cortas de unos tres segundos que ofrecieron un excelente panorama sonoro. La calidad de la grabadora mostraba un sonido de espectro completo, especialmente cuando se colocó el micrófono en la ubicación del ROV. Además de lo anterior esta grabación muestra los sonidos del ROV y del buque cuando estaban ubicados cerca de la ventila y en esta misma grabación se aprecia el peculiar *silencio* del fondo del mar.

La naturaleza del evento físico produjo sonidos constantes de baja frecuencia y, musicalmente hablando, los sonidos no ofrecen una paleta rica de transformación directa y manipulación, ya que no hay variaciones a corto plazo, pero el sonido puede ser descrito como un ruido constante con un resonador en el rango de frecuencias bajas, y la señal es interesante para la modulación de estructuras de tamaño medio y es representativa como fuente de sonido para algunas cadenas de procesos digitales.

GRABACIONES DE SONIDO EN EL BARCO THOMPSON

Otra de las actividades que desarrollé durante la expedición de Enlighten 2010 fue la realización de grabaciones de sonido en diferentes lugares a bordo del buque. Se trata de grabaciones que complementan el trabajo del fondo del mar y algunas de ellas se emplearon en la capa digital del audio de la obra final. El *Thompson* es un barco de investigación oceanográfica de 274 pies de longitud y 52.5 pies de viga y con carga completa tiene un calado de 19 pies. Sus dimensiones lo convierten en un gran resonador metálico con varias habitaciones aisladas y puertas pesadas que separan las distintas áreas y eso le permite ser un increíble generador sonoro con una gran variedad de sonidos y texturas que se pueden registrar en él. La sala de máquinas, los laboratorios científicos, la cocina, las tuberías alrededor del piso y los techos, la cubierta, los pasillos, la sala de comunicaciones, las grúas y los contenedores son algunos de los puntos que generaron gran variedad de texturas sonoras complementa-

rias las cuales formaron parte de la textura mencionada. Como se describe a continuación,⁴ dependiendo de la naturaleza del material y del sonido generado, se emplearon diferentes micrófonos para las grabaciones.



Figura 12.22. | Grabación de audio Ambisonics en la quilla del *Thompson*.

Micrófono estéreo Algunos puntos de la cubierta se grabaron con un micrófono estéreo comercial tradicional de alta calidad, que también se utilizó para los sonidos ambientales regulares. Acción: caminar por los pasillos.

Micrófono Ambisonics Muchos de los registros de la sala y un par de puntos de la cubierta fueron grabados con un micrófono Ambisonics. De este conjunto de grabaciones las realizadas dentro de la proa del buque merecen especial atención, ya que cuando el barco está en movimiento las olas golpean repetidamente la quilla y ahí se encuentra un pequeño almacén. Gracias a que la habitación está parcialmente vacía, los dispositivos metálicos que cuelgan de las paredes metálicas fueron resonadores extraordinarios, y los sonidos generados

⁴ Un video documental sobre el proceso de grabación fue gentilmente producido por el equipo de OOI: <https://youtu.be/YVJq2T9GKRI>

por cada onda que golpeaba el recipiente resultaron un hallazgo de patrones rítmicos y energía acústica. La experiencia acústica se complementó con el rápido movimiento ascendente y descendente del cuarto que, a su vez, generaba vértigo y giro. Registrar la situación en el formato de Ambisonics con la mejor calidad posible para su procesamiento y manipulación futura fue valioso e importante para el proyecto.

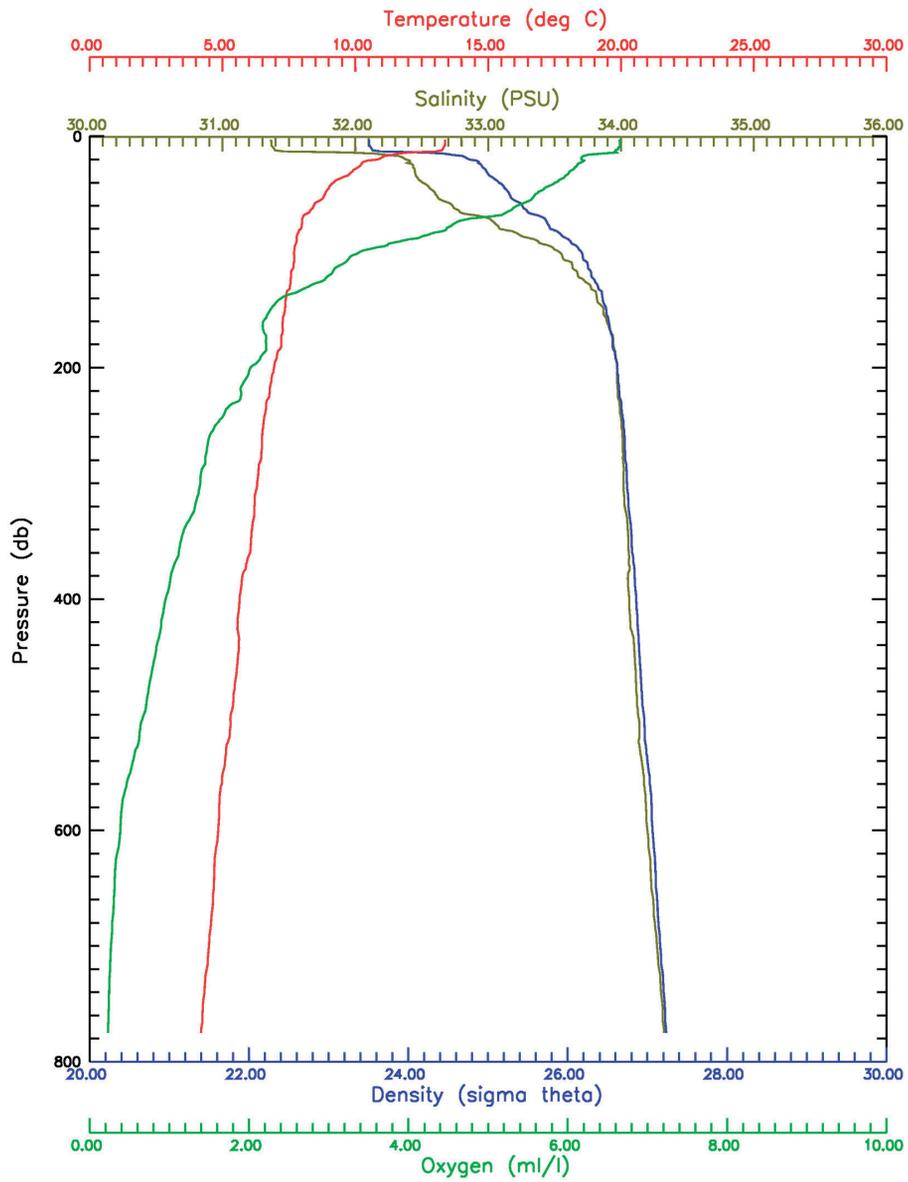
Micrófono de contacto El buque está equipado con una increíble variedad de dispositivos entre los que se encuentran equipos hidráulicos, neumáticos, eléctricos y digitales. Estos equipos se extienden por todo el buque produciendo *mundos microsónicos* que fueron grabados con micrófonos de contacto. Explorar las propiedades sonoras y las sorpresas audibles de los equipos de laboratorio también fue parte del proceso de investigación y descubrimiento de la construcción de la pieza.

Hidrófonos Las grabaciones de mar se registraron usando hidrófonos colgando del sotavento y el barlovento del barco. Estas grabaciones registraron el agua que golpea el casco y también el funcionamiento de los dos impulsos z de 3 000 caballos de fuerza que mueven el buque.

DATOS EN EL BARCO

El buque *Thompson* está equipado con dispositivos para medir una amplia variedad de información geográfica, química, física y geológica. Los rangos, la frecuencia de muestreo y la resolución del mismo, así como la variabilidad de los datos son distintos dependiendo del evento físico registrado, pero todos los datos pueden adaptarse y normalizarse a rangos sonoros que funcionaron óptimamente. El equipo científico generó una extensa cantidad de datos e información, entre los que se incluyen como ejemplo los valores CTD (conductividad, temperatura y profundidad por sus siglas en inglés), aunque solo algunos de ellos fueron utilizados para el proyecto artístico. Es importante señalar que los datos descritos en el resto del capítulo se utilizaron para modular parámetros sonoros.

Los datos consisten mayormente en un valor único que varía lentamente con el tiempo. Las pendientes y los picos son raros en los perfiles de agua donde se esperan cambios graduales lentos, pero por otro lado los mapas batimétricos ofrecen picos y valles. Los perfiles



Thomas G. Thompson

CTD cast TN252-V0

Figura 12.23. | Trazado de CTD que muestra temperatura, salinidad, densidad y cambios de oxígeno.

secundarios ofrecen imágenes multicapa de extraordinaria atracción sonora y se describirán a detalle más adelante en este mismo capítulo.

DAS

Entre los diferentes datos que los científicos produjeron durante la expedición, encontramos una colección de información registrada de equipos preinstalados en el buque. Esta colección de datos se conoce como DAS. Estas grabaciones fueron parte del registro regular de todas las expediciones. La tabla “Parámetros DAS registrados por sensores del *Thomson*” muestra todos los parámetros que están registrados. Para su representación, estos materiales fueron exportados como archivos de texto, manipulados con scripts Groovy y personalizados y leídos como buffers SuperCollider para su utilización en el contexto sonoro.

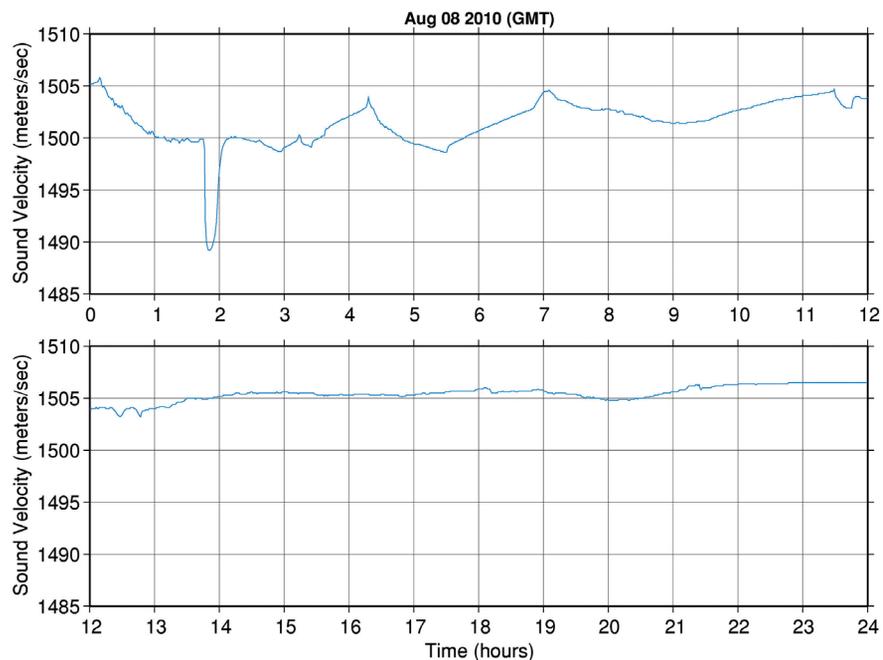


Figura 12.24. | Trazado de los cambios de velocidad del sonido en un lapso de 24 horas.

Tabla 12.2. Parámetros DAS registrados por sensores Thomson

<i>ID</i>	<i>VALUE</i>
1	Nav computer GMT date (dd/mm/yyyy)
2	Nav computer GMT time (hh:mm:ss)
3	Nav computer latitude (+/-dd.dddddd)
4	Nav computer longitude (+/-ddd.dddddd)
5	Gyro compass heading (degrees true)
6	Nav computer COG (degrees true)
7	Doppler speed log (knots)
8	Nav computer SOG (knots)
9	Thermosalinagraph sea temperature (degrees C)
10	Thermosalinagraph sea temperature external (degrees C)
11	Thermosalinagraph sea conductivity (Seimens/meter)
12	Thermosalinagraph sea salinity (PSU)
13	Thermosalinagraph chlorophyll (volts)
14	Thermosalinagraph light transmission (volts)
15	Water depth (meters)
16	IMET air temperature (degrees C)
17	IMET relative humidity (percent)
18	IMET barometric pressure (millibars)
19	PAR (microEinsteins per square meter per second)
20	IMET short wave radiation (watts/square meter)
21	Wind speed true (knots)
22	Wind direction true (degrees)
23	Wind speed relative (knots)
24	Wind direction relative (degrees)
25	Average true wind speed (knots)
26	Average true wind direction (degrees)
27	Sound velocity (meters/second)
28	Winch ID number
29	Wire out (meters)
30	Wire rate (meters/minute)
31	Wire tension (lbs.)

Batimetría

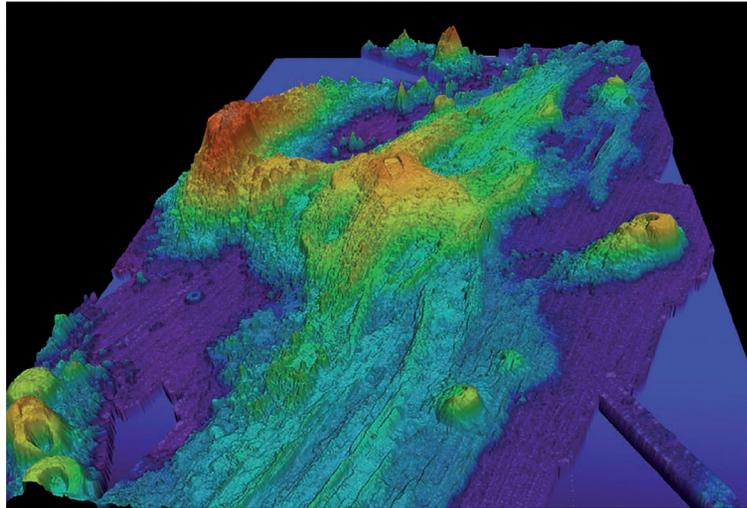


Figura 12.25. | Batimetría de la región submarina Axial.

La batimetría es la medición del fondo marino. Uno de los principales objetivos de la expedición Enlighten fue la generación de medidas batimétricas de alta resolución de la región para definir la mejor ubicación para cables y nodos del observatorio oceanográfico. Por esta razón se emplearon varios dispositivos y distintas técnicas para medir la elevación del fondo marino. La sonda de rayos múltiples EM300 montada en el casco del *Thompson* generó mapas con una resolución de 50 metros. El sonar multifunción RESON SeaBat a bordo de los ROVs Sentry y el Jason crearon mapas de resolución a escala fina.

Los diferentes datos batimétricos fueron exportados como archivos de texto y, en el caso de los datos DAS, fueron manipulados y transformados con scripts Groovy y personalizados para la alimentación de *buffers* en SuperCollider, que posteriormente modularon parámetros sonoros.

Perfiles del suelo marino

Con el *Knudsen Chirp 3260* se obtuvieron perfiles de subsuelo. El interés principal de obtener estos datos para nuestros propósitos consiste en que este material proporciona una informa-

ción multicapa, mientras que la batimetría y la información DAS ofrecen un solo valor sobre cada punto. Gracias a la multicapa los perfiles del subsuelo representan los diferentes estratos que componen el fondo marino. Por tanto, los perfiles de Knudsen pueden interpretarse como una analogía de un espectrograma sonoro.

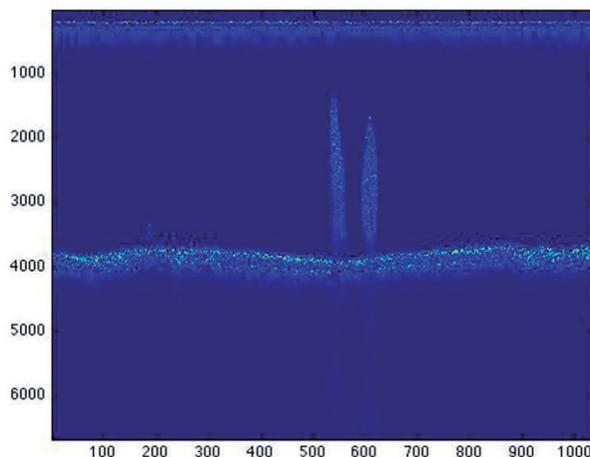


Figura 12.26. | Perfiles del subsuelo marino en donde se observan dos chimeneas.

SÍNTESIS DE AUDIO EN LA NUBE

Síntesis de audio en la nube (Cloud Synthesis) es el nombre que decidí para referirme a la ejecución de síntesis de sonido *offline* y al procesamiento, utilizando la infraestructura de cómputo en la nube.⁵

Para realizar dicha síntesis decidí crear una infraestructura de renderización distribuida (más por curiosidad que por necesidad tecnológica) y, luego, sabiendo que una cantidad significativa de datos sería producida por el equipo oceanográfico vi la oportunidad de aplicar los conocimientos para generar una solución original en la nube. Con estos antecedentes

⁵ Al parecer, dentro de la comunidad de artistas sonoros la velocidad de cómputo en las computadoras actuales es suficiente tomando en cuenta que, hasta hace algunos años, era normal esperar largos periodos para hacer el render de una manipulación sonora. Sin embargo, está abierta la pregunta respecto a cómo impactará mayor capacidad de procesamiento digital al proceso de creación sonora. En particular, en el campo de la sonificación y la minería de datos, así como en el cómputo distribuido y en la nube son áreas de experimentación.

tuve que trabajar en una infraestructura digital que aprovechara las ventajas de Amazon Web Services (<http://aws.amazon.com/>), un sistema que permite la generación de escenas de sonido mediante un servidor de SuperCollider corriendo en una instancia de Amazon.⁶

FORMALIZACIÓN DE LA OBRA

Como se puede apreciar, la preparación, el viaje conceptual y la materialización del proyecto tomaron una cantidad significativa de tiempo. Se exploraron y desarrollaron diversas ideas y lógicas artísticas para generar el trabajo, pero todas estas ideas, bocetos, pruebas y preparativos fueron de extrema importancia para llegar a la conclusión y al recorrido de la pieza final.⁷ La última iteración de la idea que se convirtió en la obra de arte se explica a continuación.

Explicar el proceso de composición es para mí, sin duda, el paso más difícil para aclarar el proceso de desarrollo de *Axial*. Explicar un proceso compositivo requiere un autoanálisis y una autorreflexión difícil y compleja. La introspección para poder explicar cómo ha sido creada una pieza es complicada, pero puede ser útil para evidenciar un proceso creativo.

PROCESO DE COMPOSICIÓN

Como ya hemos señalado, las ideas, imágenes, conceptos y elementos del proyecto tuvieron que materializarse en una pieza de arte formal que pudiera abarcar poéticamente la lógica deseada. Se concibieron y propusieron varios formatos y aunque algunos fueron descartados en el camino, después de discusiones y observaciones con colegas y artistas, la pieza llegó a su conclusión óptima gracias a los recorridos previamente trazados.

Después de meses de planificación, la pieza fue conformada y materializada en la iteración final. Se seleccionaron materiales específicos que incluían tecnologías, código y hardware diseñados específicamente para la experiencia sonora y, en otras palabras, estaba llegando al

⁶ Dado que el sistema de composición en la nube se ha utilizado y se continúa madurando, existe la intención de hablar sobre este tema en un futuro libro.

⁷ Me gustaría que los estudiantes de arte digital pudieran aprovechar esta experiencia para asimilar que algunos proyectos pueden requerir mucho tiempo para su producción óptima.

momento crucial en el que debía tomar decisiones para lograr una pieza concluida. ¿Cómo fue el proceso de composición? ¿Cómo se formó la pieza en su manifestación final? Para clarificar este punto revisemos un conjunto de hechos en el tiempo:

- La idea de trabajar con el mar como proyecto final se concibió en 2008, mientras vivía en Puget Sound.
- En 2009 asistí a la expedición Lumens.
- En 2010 participé en la expedición Enlighten y recopilé los materiales necesarios donde varias exploraciones se propusieron e iniciaron, pero nunca se solidificaron en un trabajo de acabado.
- En 2011 produje los módulos de control plurifocal, descritos en un total de 18, mientras realizaba una residencia de arte en Tokio, Japón; y un par de trabajos de ese periodo influyeron en la propuesta de *Axial* para darle un nuevo enfoque al proyecto. Durante este tiempo la idea de trabajar con un componente mecánico ganó muchísima fuerza y se crearon dos obras de arte con principios similares como preparación y prueba para *Axial*.
- En agosto de 2011 ya todo estaba configurado y se había producido suficiente exploración alrededor del concepto principal de la pieza y, aunque ninguna de las actividades anteriores fue conscientemente organizada o planeada en específico para la materialización de la pieza tal como fue producida, en última instancia fueron gestores definitorios de la obra final. Hasta ese momento había un conjunto de oportunidades, exploraciones heurísticas, curiosidad e investigación informal que evolucionaban gradualmente hacia la obra concluida. Esta combinación fue una acumulación de deseos sonoros intangibles y una serie de experimentos previos que devenían obras de pequeño formato pero que, para ese momento en el tiempo, definían la pieza que tenía que ser compuesta y exhibida.

La recolección de ideas, descartando algunas de ellas y modificando otras, así como confrontando los beneficios y los problemas de cada una de ellas fue un proceso natural realizado a lo largo de varias iteraciones. Realizar bocetos y dibujos ha sido una manera de

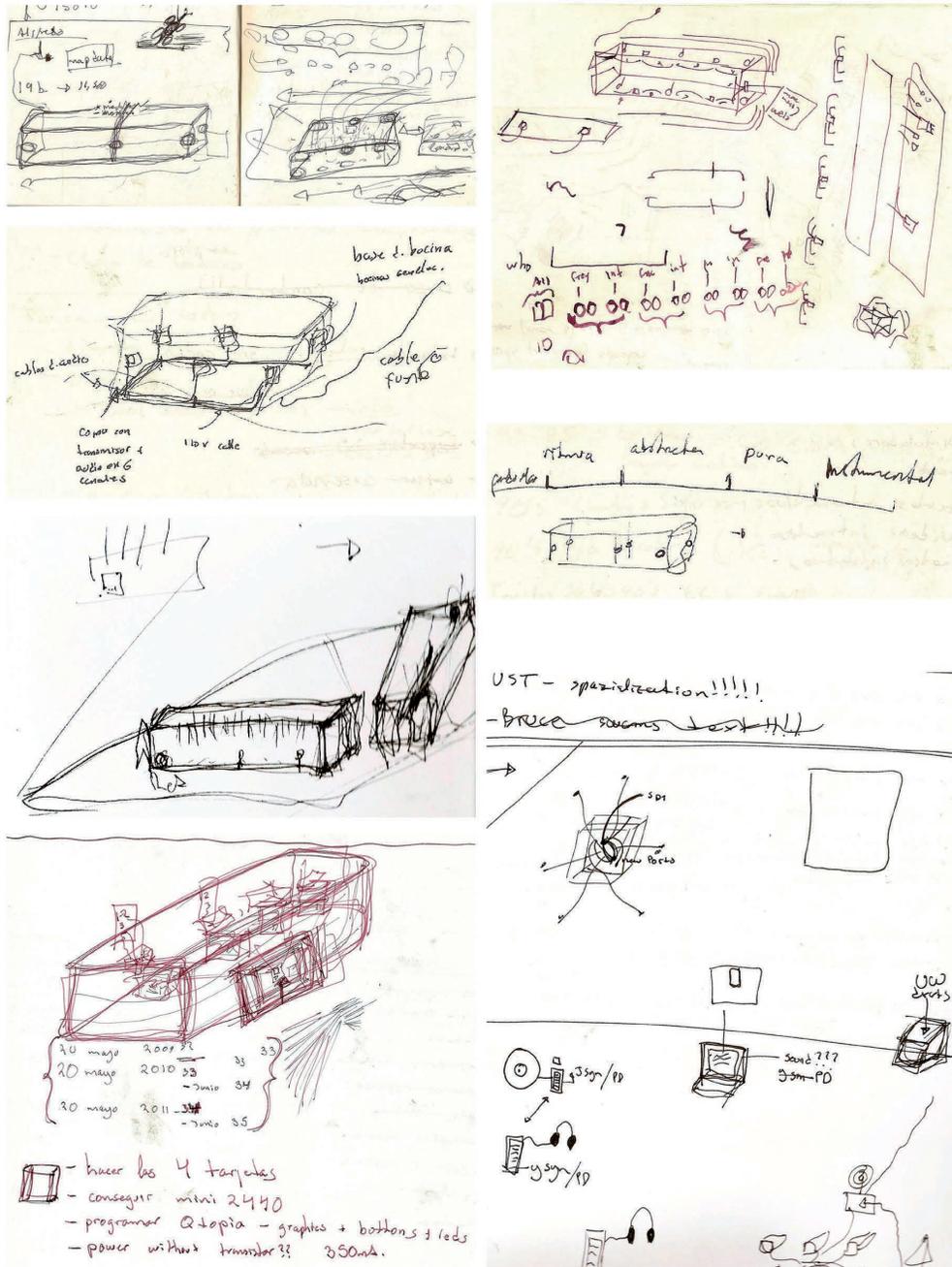


Figura 12.27. | Bocetos de diferentes ideas para la instalación.

materializar pensamientos (véase la figura 12.27). Aunque el componente sonoro y conceptual fue una guía para la toma de decisiones artísticas, el camino que proseguí fue una mezcla de compromisos artísticos, logísticos y técnicos. Con una técnica de evaluación heurística, la pieza fue gradualmente autogestionada y después de tres meses alcanzó su estado final.

Planificación de la manifestación física

Los puntos de partida para materializar la pieza fueron bocetos en un cuaderno de dibujo en el que imaginaba posibilidades físicas y distribución de elementos. Una limitación importante consistía en saber con antelación la ubicación posible para la instalación de la obra. Haber alquilado un contenedor de 10 pies y haber realizado un conjunto de experimentos y grabaciones un año antes ayudó a concebir la calidad del sonido y las proporciones de la experiencia. A partir de esas pruebas yo ya sabía que el contenedor sería un resonador extraordinario y para esta etapa del proceso creativo me era muy evidente que, para aprovechar las propiedades sonoras del contenedor, colocaría un gran número de solenoides a lo largo de las paredes del contenedor. Así, partiendo de esa lógica, el punto de atención estaba en el desarrollo sonoro.

Proceso compositivo del material sonoro

La pregunta más profunda en cuanto a todo el proyecto era cómo debía sonar en materia de resultado sonoro. El sonido tendría que ser la guía y el elemento central y debía transmitir la lógica y la esencia de la experiencia previa en el mar. Mientras que la estética había sido parcialmente establecida desde el principio siguiendo la conceptualización, el diseño, la producción y la realización de trabajos anteriores, cuando se inició este proyecto no tenía una idea definida acerca del resultado de la experiencia sonora. Una vez más, un método de *lluvia de ideas* y un conjunto de decisiones iterativas a lo largo de varios meses le dieron forma a la experiencia sonora ideal.

Mi punto de partida fue la recolección de grabaciones sonoras durante las expediciones en el barco, específicamente, las grabación de chimeneas de las ventilas hidrotermales. Pero,

además, yo ya tenía un conjunto de códigos SuperCollider desarrollados durante la segunda expedición que permitían utilizar los datos científicos generados por el equipo de científicos como moduladores de parámetros sonoros.

Para este momento, ya había terminado un conjunto de obras que empleaban dispositivos mecánicos y que generaban sonidos acústicos con controles digitales y, por tanto, me fue muy natural continuar con tal exploración de texturas de sonido acústico. El hardware y el software estaban listos para ser utilizados para el proyecto y podría ser la materialización de una investigación a largo plazo gestada durante varios años.

Algunas de las piezas mecánicas anteriores ya tenían un componente interactivo que permitía la modificación de estructuras y elementos sonoros, dependiendo de la actividad física y el comportamiento de los participantes y, si bien esos proyectos funcionan como experiencias interactivas con el público, para esta ocasión quería producir una pieza que no dependiera de la interactividad directa. Mi reto fue producir un trabajo lineal no interactivo atractivo y motivador, con lo cual, para este contexto, decidí trabajar en una pieza de un bucle fijo compuesto por dos capas, un material acústico producido con una plataforma mecánica y material digital compuesto por el sonido y las sonificaciones obtenidas en el barco.

La duración y la estructura de la obra no se determinaron desde el principio, sino que fueron definiéndose gradualmente mediante un proceso iterativo (intentar, probar, redefinir y reorganizar). El software Juum, que se explica en el capítulo 3 “Dúo Juum”, me permitió una fácil manipulación y una reorganización de estructuras, mientras que el Plurifocal Events Controller, que se explica en el capítulo 18 “Plurifocal sound controller”, me permitió la pro-

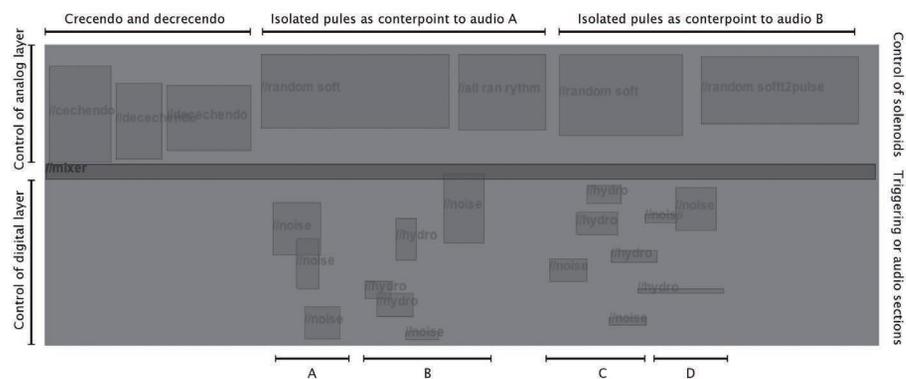


Figura 12.28. | Partitura de *Axial* mostrando capas y secciones.

ducción de patrones rítmicos distribuidos. Para esto, es de destacar que tales patrones pueden ser activados directamente desde la línea de tiempo de Juum, hecho que permitió que el proceso de exploración fuera relativamente simple.

Proceso compositivo de la capa analógica

A pesar de que las capas (analógica y digital) se produjeron al mismo tiempo y con influencia recíproca entre sí, la analógica fue la detonadora de la estructura y de la duración de la obra.

Como se explica en el capítulo 18, “Plurifocal sound controller”, la última versión del Plurifocal Events Controller es un controlador que permite la creación y la manipulación de estructuras polirrítmicas repartidas en múltiples fuentes y ubicaciones. Las primeras exploraciones me condujeron a estructuras complejas y texturas intrincadas que se asemejan a algunas composiciones de percusión; sin embargo, después de un periodo inicial de emoción y sorpresa, el gesto *musical* comenzó a sentirse fuera de lugar. Con este hallazgo rítmico comencé a sentir –o escuchar, debo decir– que la lógica musical tradicional estaba interfiriendo más que apoyando en la lógica sonora de la pieza, ya que después de todo este trabajo no fue concebido como una composición musical, sino como una experiencia sonora con otros tipos de relaciones entre el tiempo, el sonido, los oyentes y el espacio físico. En este instante y –más que nunca– sentí y viví la diferencia entre la tradición musical y el concepto del arte sonoro.

Con cada iteración del proceso compositivo la complejidad polirrítmica inicial se hizo más delgada y más directa. Ante este momento de gestación-desarrollo de la pieza viví un proceso de decantación que surgió como un mecanismo para homogeneizar la idea central de la misma. La complejidad se transformó en un solo gesto que tradujo la lógica idónea de los procesos oceanográficos. El periodo de tiempo de los procesos del océano es extremadamente largo y las transformaciones son lentas y graduales, pero, inevitablemente notables a largo plazo. La lógica del sonido debía revelar esa lógica y develar la sensación de altamar previamente experimentada.

Entre los primeros esbozos y la versión final de la obra pasó casi un año, periodo durante el cual se produjo un acontecimiento dramático que ayudó a solidificar la estructura compositiva de la pieza. En abril de 2011 hizo erupción el volcán Axial, cumpliéndose la

predicción de que esto ocurriría antes de 2014. El evento fue cuidadosamente seguido por la comunidad oceanográfica y extremadamente importante para los científicos de la Ocean Observatory Initiative, ya que el volcán Axial se encuentra en el área de estudio del observatorio. La erupción no fue monitoreada y la oportunidad de registrar el evento se perdió debido a que el equipo de la OOI para ese momento aún no estaba instalado.

La erupción *Axial* de abril de 2011 es una demostración de la importancia de transmitir información sísmica, video y de fluidos en tiempo real. El evento eruptivo, o serie de eventos, durante el mes de abril de 2011, no fue detectado hasta que una serie de cruceros ROV/AUV, planeados por separado y con años de antelación, hicieron observaciones para descubrir, confirmar y detallar la transformación de *Axial* (Chadwick *et al.*). Lo que aconteció durante y poco después de la erupción permanecerá en gran medida desconocido, ya que los flujos extraordinarios iniciales de calor, química y biología se han deteriorado en los tres meses y medio que han transcurrido (Proskurowski, 2011).⁸

Para este momento yo ya estaba trabajando en piezas que utilizaban dispositivos mecánicos para producir grandes masas de sonido. Imaginar la acumulación gradual de energía y la liberación del volcán desencadenó lo que se convertiría en la estructura y el gesto final de la obra. Después de la erupción, visualicé el gesto sonoro en su totalidad: un trazo directo y mínimo como mecanismo para unir las diferentes secciones que constituyen la obra. La erupción no registrada del Axial se convirtió en la estructura narrativa de la pieza y, tras tomar esa decisión, el resto del proceso compositivo fluyó orgánicamente.

Durante varios días jugué con diferentes algoritmos que generarían y controlarían los patrones rítmicos que los módulos plurifocales habían de producir. Una vez más un proceso iterativo de probar-escuchar-modificar me llevó a la célula rítmica que genera la acumulación gradual. Los patrones y la duración de cada evento fueron controlados gráficamente con la línea de tiempo del software Juum. El siguiente código pertenece al elemento *crescendo* de la partitura de Juum.

```
1. //crescendo element
2. /**
3. //automator script for plurifocal TWS
4. //plurifocal
5. //      you can use sendMSG(node)
```

⁸ Traducción del autor.

```
6. //nodes
7. //     it is an array of nodes with
8. //     location.x
9. //     location.x
10. //     location.x
11. //     value
12. //     active
13. //     id
14. //     netID
15. //     range
16. //     parameters
17. //serial
18. //     you can use serialOut(String)
19. //active
20. //     it is a checkbox
21. //sliders
22. //     it is an array of JSlider
23. //buttons
24. //     it is an array of JButton
25. // *
26.
27. //GLOBALS
28. address = 65535
29.
30. rand = new Random()
31.
32. boardsIDs = [18, 14, 7, 13, 8, 12, 22, 23, 27, 24,
33. 29, 20, 30, 33, 32, 31, 39, 34, 36, 38
34. ]
35.
36. //ACTIONS
37. void boardsSet(intensity, period){
38.   String intS = Integer.toString(intensity, 16)
39.   String perS = Integer.toString(period, 16)
40.   serial.serialOut("$address FF 00 $intS $perS N\n\r»)
41. }
42.
43. void motorSetAll(board, motor, intensity, period){
44.   String brdS = Integer.toString(boardsIDs[board], 16)
45.   String mtrS = Integer.toString(motor, 16)
46.   String intS = Integer.toString(intensity, 16)
47.   String perS = Integer.toString(period, 16)
48.   serial.serialOut("$address $brdS 01 $mtrS $intS $perS N\n\r»)
49. }
50.
51. void motorSetIntensity(board, motor, intensity){
52.   String brdS = Integer.toString(boardsIDs[board], 16)
53.   String mtrS = Integer.toString(motor, 16)
54.   String intS = Integer.toString(intensity, 16)
55.   serial.serialOut("$address $brdS 02 $mtrS $intS N\n\r»)
56. }
```

```

57.
58. void motorSetPeriod(board, motor, period){
59.   String brdS = Integer.toString(boardsIDs[board], 16)
60.   String mtrS = Integer.toString(motor, 16)
61.   String perS = Integer.toString(period, 16)
62.   serial.println("$address $brdS 03 $mtrS $perS N\n\r»)
63. }
64.
65. void boardsSync(){
66.   serial.println("$address FF 04 N\n\r»)
67. }
68.
69. void boardsOff(){
70.   serial.println("$address FF 05 N\n\r»)
71. }
72.
73. def start(){
74.   //boardsOff()
75. }
76.
77. def stop(){
78.   //boardsOff()
79. }
80.
81. pause = 10 * 5
82. counter = 0
83.
84. def loop(){
85.   if(counter++ % pause == 0){
86.     boardsSet((int)(currentPos * 100), 7)
87.   }
88. }

```

Resulta evidente la posibilidad de ver que las variables globales se refieren a los módulos ID. También hay un conjunto de funciones (no utilizadas en este elemento en particular) que permiten el control de los motores mediante el ajuste de la intensidad, periodicidad o sincronización. Para este elemento en particular, la función de bucle incrementa gradualmente la intensidad de todos los solenoides según la variable *currentPos*. (Para una descripción detallada del control de los módulos consulte el capítulo 18, “Plurifocal sound controller”.)

Una vez alcanzada la idea general de la pieza y la fase temporal del trabajo obtenido, la parte analógica con motores se complementó con el material digital.

Proceso compositivo de la capa digital

Como describí al principio del capítulo, durante la expedición a Axial en agosto de 2011 desarrollé un conjunto de scripts de SuperCollider que permitirían la transformación de grabaciones de sonido y un conjunto de códigos que también permiten la modulación de sonido, utilizando los datos recogidos por el equipo de científicos. Con estas técnicas realicé un conjunto de piezas de sonido de corta duración (creadas en un ordenador personal a bordo del buque *Thompson*), y aunque estos archivos de audio deben considerarse pruebas y exploraciones más que una composición terminada, generaron un conjunto de texturas sonoras interesantes y valiosas. Entre noviembre y diciembre de 2011 terminé la arquitectura para la renderización de sonido en Amazon Cloud, permitiendo la producción de texturas más largas y utilizando un conjunto más grande de archivos de datos.

El código Groovy permite la generación de una partitura osc para SuperCollider de un archivo de Knudsen. Para simplificar, el código se presenta en su versión para computadora personal y resulta legible un archivo de perfil de subfondo generado por el Chirp de Knudsen, y mapea la densidad de cada punto físico en un grano de sonido. Con este código y la posterior representación de la partitura generada se produce una textura de síntesis granular en la que densidad y frecuencia de grano se modulan con base en la información del perfil del subsuelo marino. De esta manera, los datos geológicos se utilizaron directamente para modular los parámetros sonoros de una sección de sonido. Las propiedades del perfil resultan audibles y se pueden escuchar claramente y apreciar sonoramente en su totalidad.

```
1. /*this script generates a sc score  
2. based on a file generated by matlab  
3. and following parameters */  
4.  
5. inputfile = args[0]  
6. outputfile = args[1]  
7. duration = args[2].toDouble()  
8. volReduction = args[3].toInteger()  
9. overlap = args[4].toInteger()  
10.  
11. buildScore(inputfile, outputfile)  
12.  
13. def buildScore(inputfile, outputfile){  
14.   input = new File(inputfile)  
15.
```

```
16. //first we count the number of lines
17. lines = 0;
18. input.withReader{reader ->
19.   reader.eachLine{currentLine ->
20.     lines++
21.   }
22. }
23.
24. println lines + " amount of lines"
25.
26. durStep = duration / lines
27. durGrain = durStep * overlap
28.
29. out = new File(outputfile)
30. out.write("") //to override
31. out << "[\n"
32. idCounter = 0
33. lineCounter = 0
34. location = 0
35. input.withReader{reader ->
36.   reader.eachLine{currentLine ->
37.     elements = currentLine.split(",")
38.     numElements = elements.size() / 2
39.
40.     location = lineCounter * durStep
41.     dur = durGrain
42.     numElements.times{elementCounter ->
43.       id = 1000 + idCounter++
44.       freqRaw = elements[elementCounter * 2].toDouble()
45.       freqScale = freqRaw * 0.5 + 0.3
46.       freq = Math.pow(2, freqScale * 10) * 20
47.       volRaw = elements[elementCounter * 2 + 1].toDouble()
48.       dbRaw = Math.pow(10, volRaw / 20)
49.       dbScaled = dbRaw - volReduction
50.       vol = Math.pow(10, dbScaled / 20)
51.       ran = new Random()
52.       minimalMove = ran.nextDouble() * 0.05
53.       elementLocation = location // + minimalMove
54.       out << "[$elementLocation, ['s_new', 'sinusoidal', $id," +
55.         "0, 0, 'dur', $dur, 'freq', $freq, 'vol', $vol] ],\n"
56.     }
57.     lineCounter++
58.   }
59. }
60. finalDuration = location + 10
61. out << "[$finalDuration, ['c_set', 0, 0] ]\n"
62. out << "]\n"
63. println "Done writing score"
64. }
```

La colección de sonido generado fue un punto de partida natural para darle forma a la capa digital de la instalación sonora. Por tanto, una vez que se definió la estructura global de la pieza y se bosquejó la parte analógica, ésta se complementó con los materiales sonoros. Una vez más la línea de tiempo en el programa Juum ayudó a organizar los sonidos mediante la manipulación de los volúmenes y distribuciones Ambisonics,⁹ así como los puntos de inicio y final. La organización de los materiales se produjo de forma similar al resto de la composición planificando distribuciones de sonido heurístico, escuchando y reajustando gradualmente los elementos hasta alcanzar un sentido de equilibrio con un diálogo atractivo entre las partes analógicas y digitales. Mediante sucesivas aproximaciones las texturas sonoras digitales se dispusieron generando un contrapunto con los impulsos mecánicos.

El proceso compositivo entre los meses de abril a noviembre de 2011 fue apoyado por una emulación digital de la parte mecánica y por un sistema de sonido cuadrafónico. La calibración final no se pudo realizar hasta que obtuve el contenedor físico; a finales de noviembre de 2011 alquilé el contenedor y me ubiqué en el lugar de exhibición.

Proceso de composición durante la instalación, ejecución y calibración

Del 2 al 7 de noviembre de 2011 instalé los componentes mecánicos en el contenedor. La decisión sobre dónde y cómo montar los solenoides la tomé hasta que el proceso de instalación del contenedor había comenzado, y el hecho de estar físicamente dentro del contenedor me dio un sentido real sobre la distribución del sonido y su resonancia. Una vez *in situ* decidí distribuir uniformemente los solenoides a nivel del oído y colgar los altavoces en las esquinas superiores del contenedor.

Una vez completada la instalación comenzó un periodo de calibración en el que se tomaron algunas decisiones de composición adicionales en relación con la acústica específica del contenedor a los cambios perceptivos de estar en un contenedor oscuro y vacío. Esta particularidad abrió una pequeña ventana para la calibración y los ajustes de composición finales. Los cambios más significativos ocurrieron en los niveles de amplitud de las diferentes capas y en los elementos sonoros. La revisión final de las distribuciones del tiempo, las

⁹ Es importante recordar que Juum controla un servidor SuperCollider para la generación de sonido.

velocidades del patrón y las igualaciones también ocurrieron en sitio en los días anteriores a la exhibición de la pieza al público.

AXIAL ESTRUCTURA Y PARTES

Manifestación física

La pieza se materializa en un contenedor de carga de 40 pies. El contenedor es el límite del micromundo que ha sido creado como una referencia al mar y, debido a su oscuridad, resulta análogo al mar profundo además de generar una frontera clara entre el interior y el exterior que nos permite contar con un sitio oscuro y no es, únicamente, el espacio donde la pieza existe, sino que también se convierte en el espacio sonoro en sí mismo. El espacio es el resonador y genera un conjunto de relaciones con el exterior que interactúan con la audiencia que se encuentra dentro del contenedor y con el contenedor en sí mismo ubicado en el espacio. Por otra parte, el contenedor es igualmente la metáfora del barco alojado en el vacío del océano exterior.

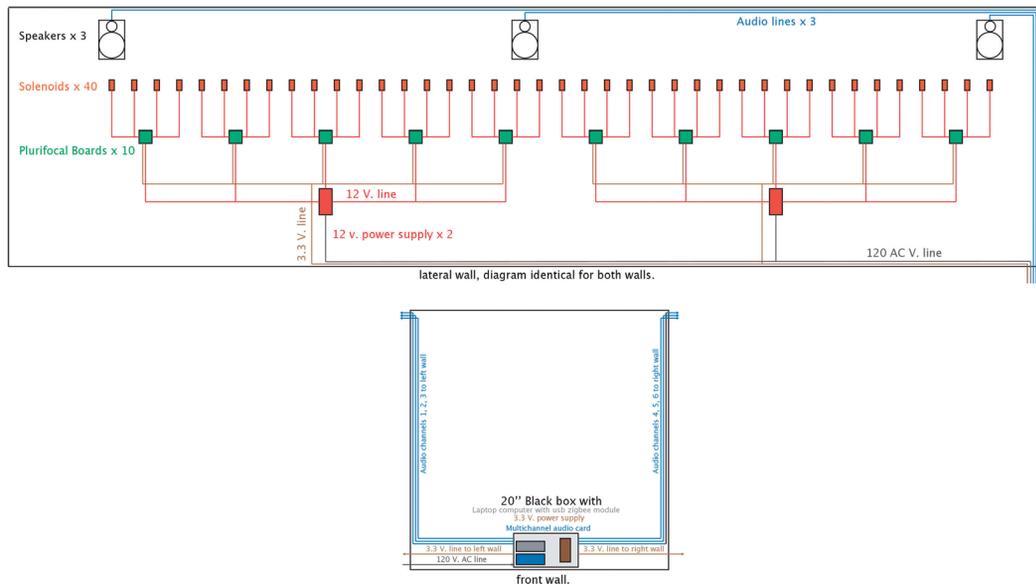


Figura 12.29. | Diagrama de conexiones y partes, paredes frontal y laterales.

Aunque no totalmente formalizada antes de la exposición, la pieza genera una conversación dual según la ubicación del público. Se puede apreciar desde el interior, generando una experiencia sonora inmersiva; o puede ser apreciada desde el exterior, convirtiendo al público en un observador que *encuentra* un cuerpo sonoro *plantado* en el medio del vacío.

Estructura composicional

Como ya hemos comentado, la organización del material sonoro no siguió una composición formal o una estructura en el sentido de la forma musical tradicional. El trabajo fue intuitivo y se basó en intentar, escuchar, modificar y afinar longitudes, partes y la superposición de elementos audibles. La composición tuvo que interrelacionarse y crear un diálogo y el contrapunto entre dos tipos de materiales: la capa digital y la capa acústica de sonidos de percusión producidos con dispositivos mecánicos. La figura 12.30 presenta las secciones de ambos materiales y ofrece una mejor representación de la estructura composicional.

La capa digital comienza con una sección creada con base en material de sonificación de datos seguido por la transformación de una grabación del buque. Posteriormente, una sección hidrotérmica de alta frecuencia anticipa una sección larga realizada mediante grabaciones de hidrófono que contienen sonidos del Jason ROV. Después de esto la grabación del hidrófono, que tiene interferencia eléctrica, anticipa una última versión de la misma la cual funciona como refuerzo del componente de baja frecuencia. La lógica es simple y genera periodos medianos de tensión y calma.

El material análogo está creado con base en tres secciones separadas sobre la pieza. La primera intervención mecánica se realiza con pulsos regulares y fijos. La segunda parte está concebida como patrón rítmico básico que crea dos elementos de *crescendo* y *decrecendo*, pero aun cuando en esta parte el patrón rítmico es fijo, tiene cierta inestabilidad que soporta el diálogo acústico con la cuarta parte de la capa digital. La última sección analógica constituye la sección más alta y más agresiva de toda la pieza porque es la sección que simula la erupción volcánica y consiste en un *crescendo* a largo plazo de todos los solenoides sincronizados. El nivel acústico (debido a la resonancia del recipiente) es extremadamente fuerte durante esta sección climática de la obra.

La pieza tiene una duración de 30 minutos y 44 segundos en *loop* indefinido durante la presentación de la instalación sonora. Dado que la gente es libre de entrar y salir del contenedor en cualquier momento, la lógica de la pieza no se basa en una evolución temporal de las ideas sonoras porque la composición sonora está ligada a las relaciones espaciales donde la organización del tiempo es secundaria, producto de la relación espacial con el oyente. La composición fue creada con el objetivo de generar un conjunto de diferentes experiencias sonoras con transiciones lógicas. De esta manera la superposición entre capas y las transiciones entre secciones tiende a ser lisa con largos periodos de fundido cruzado y evoluciones graduales.

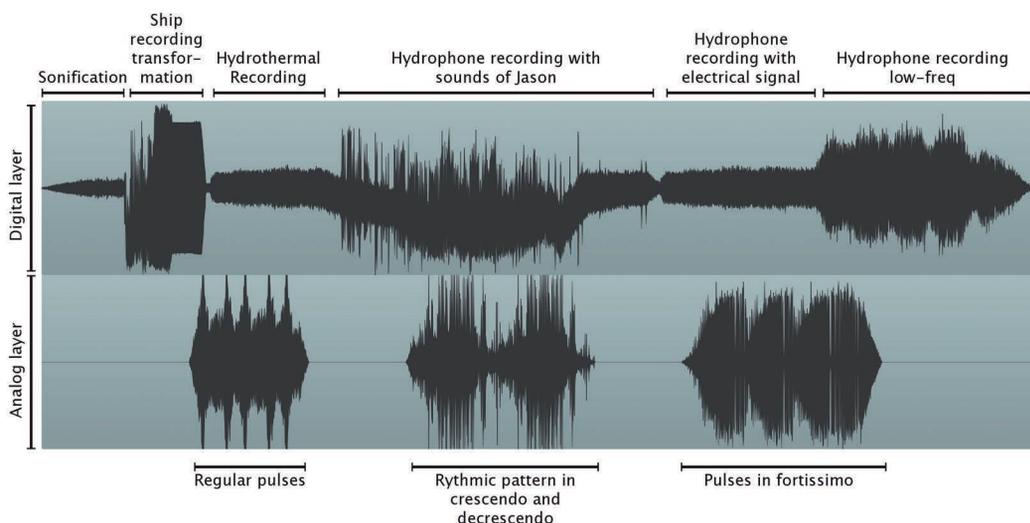


Figura 12.30. | Partes y secciones de un audio *loop* de 30 minutos.

MONTAJE

Ubicación

Hasta el momento, *Axial* ha sido experimentado por el público una única vez, durante el invierno de 2011. Esto no es una obra de arte para espacio específico, ya que podría existir en diferentes lugares y situaciones, pero el tamaño del contenedor necesariamente genera una relación entre el objeto y el lugar donde se encuentre. Esta relación se expresa en dos aspectos: la interacción física entre el contenedor y el lugar, y el significado metafórico de



Figura 12.31. | Vista del contenedor localizado en el Center for Urban Horticulture en la UW. (Fotografía de Daniel Hawkins.)

contar con un contenedor de carga en una ubicación particular. Cuando se decidió que el formato de salida de la obra sería en tierra (dentro de un contenedor de carga en lugar de ser transmitido en vivo desde el mar), como se describió y detalló anteriormente, era necesario decidir el lugar específico de exhibición. Para tomar la decisión de la ubicación del contenedor se consideraron varios lugares, entre ellos:

- La bodega Fremont DXARTS
- Muelle de carga y transporte del Departamento de Oceanografía de la UW
- Área marítima Harbor Island
- Arboreto de Washington Park
- Elliott Bay Waterfront
- Center for Urban Horticulture en la UW

La idea principal concebía seleccionar un lugar en un espacio abierto lo suficientemente grande como para emplazar un contenedor de carga de 40 pies de tamaño completo, donde el público pudiera tener la sensación de estar rodeado por un espacio abierto que lo refiriera a altamar, donde el referente espacial fuera lo suficientemente grande como para percibir

el contenedor pequeño a la distancia y poder acercarse al objeto desde diferentes puntos. El espacio debía ser lo suficientemente abierto como para tener una vista al horizonte y, además, debía ser silencioso. Considerando que el espacio debía funcionar como una reminiscencia y como una analogía del océano y en él un barco, el Center for Urban Horticulture en la UW resultó idóneo.

Cada uno de los lugares antes mencionados podría haber generado valiosas relaciones de curaduría con la obra de arte, y cada ubicación ofrecía problemas logísticos para ser resueltos, pero después de una negociación natural entre factibilidad, propiedades geográficas y el significado de la ubicación se seleccionó el Centro de Horticultura Urbana.

El Center for Urban Horticulture forma parte de los jardines botánicos de la Universidad de Washington, incluye 16 acres de jardines y 74 acres en la reserva natural de Union Bay;

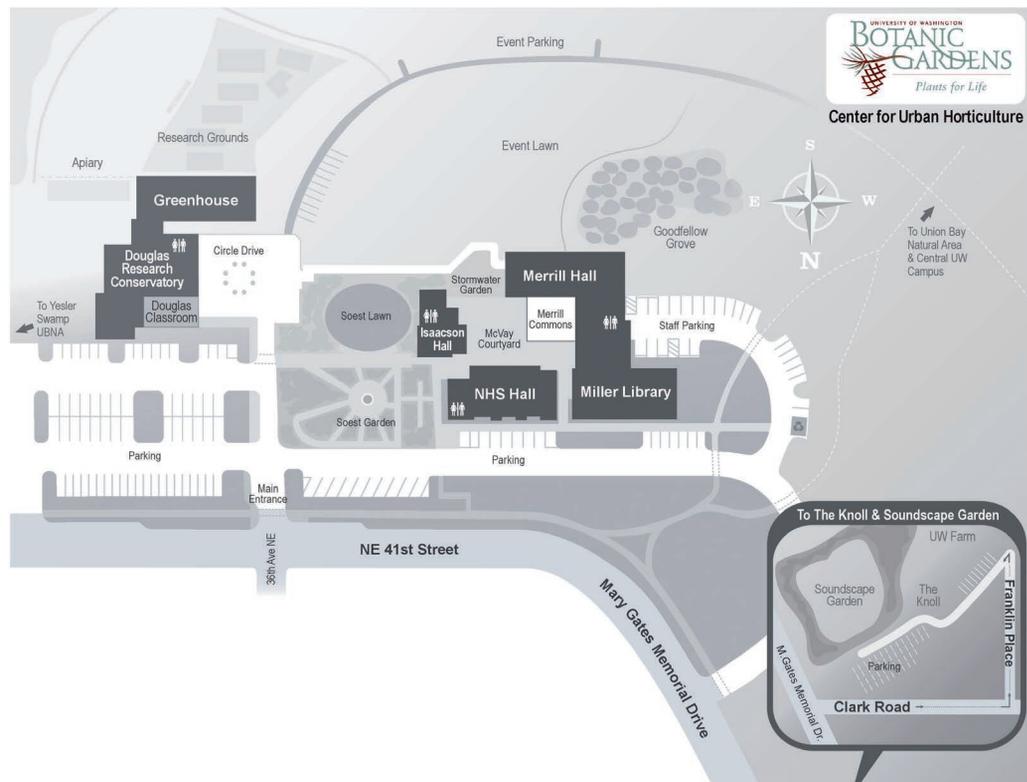


Figura 12.32. | Mapa del Center for Urban Horticulture de la Universidad de Washington.

fue en esta región donde se seleccionó el lugar específico y la ubicación exacta donde se colocó el contenedor fue en la latitud 47.659051 y longitud -122.291855, colindando al norte con el edificio de la UW Ceramic Metal Arts; al sur con las tierras de cultivo para la investigación botánica; al oeste con la entrada a la reserva natural de Union Bay; y al este con la avenida Mary Gates.

El lugar resultó el ideal por varias razones: pertenece a una instancia académica que entendía la naturaleza del proyecto y se encuentra en un espacio abierto y silencioso. Además de lo anterior, está ubicado en una pequeña colina que permite una vista al horizonte sin ser obstruida por edificios.

Configuración

El contenedor de carga fue alquilado de la Aztec Containers Company ubicada en Rochester, Washington (<http://www.azteccontainer.com>). La compañía trasladó el contenedor en camión y lo colocó en el lugar y en la dirección correctos, orientando las puertas a la vista de Union Bay. Más tarde, con el apoyo de un equipo de voluntarios, instalé la pieza durante un periodo de cuatro días. Primero se limpió el contenedor y se colocaron los ocho altavoces al nivel del oído con el cable aéreo. Posteriormente, ubicamos los módulos de control pluri-focal y las fuentes de alimentación se colgaron con cables. Finalmente, el conjunto de 80 mecanismos de percusión mecánica se colocaron al mismo nivel (distribuido uniformemente sobre tres caras interiores del recipiente). Al final de la instalación de los dispositivos éstos se conectaron a las fuentes de alimentación. La energía eléctrica fue llevada al contenedor por un cable de extensión de potencia procedente del edificio de arte.

Dentro de una pequeña caja negra al final del contenedor se colocó una computadora portátil, se trata de la computadora responsable de ejecutar el software Juum con la composición de *Axial*.

Presentación

El trabajo de la obra final fue concebido para ser experimentado dentro del contenedor y percibir la mezcla entre el material análogo, los sonidos digitales y las resonancias del espa-



Figura 12.33. | Solenoides montados en la pared del contenedor.

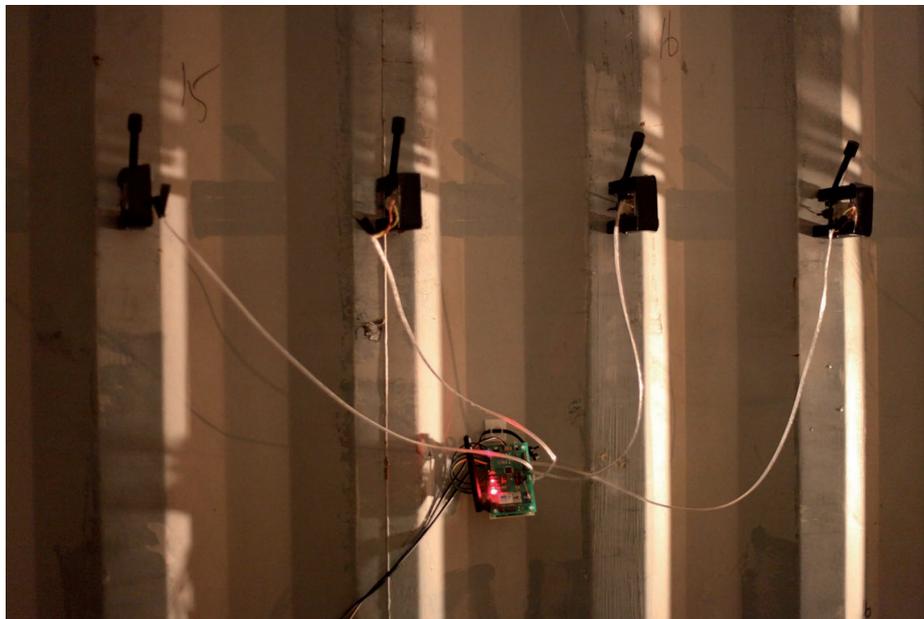


Figura 12.34. | Grupo de cuatro solenoides y un Plurifocal Events Controller.

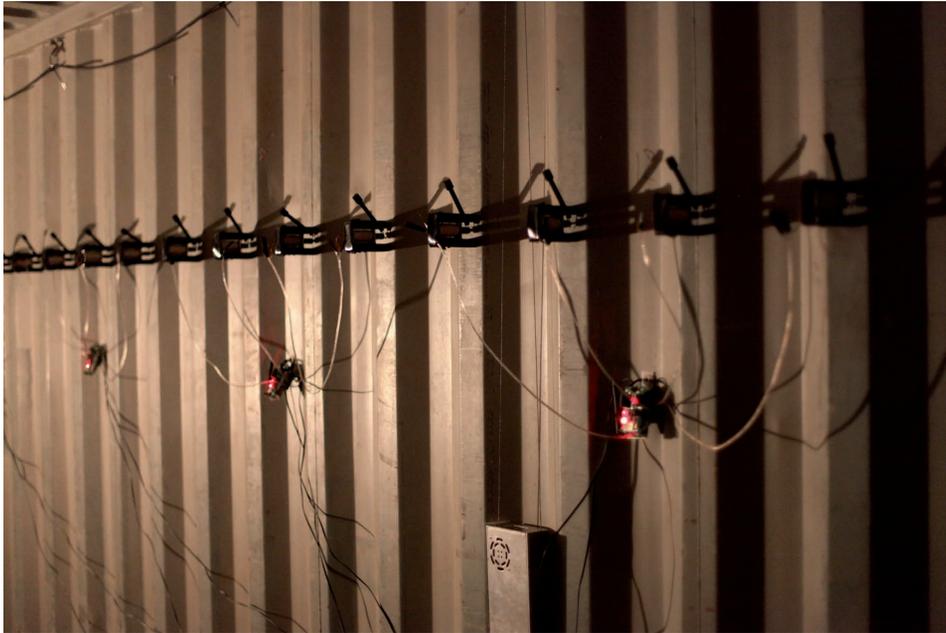


Figura 12.35. | Pared de solenoides, PEC y fuente de poder.



Figura 12.36. | Configuración del contenedor.

cio. El trabajo se pudo apreciar desde el atardecer hasta la medianoche, creando una sutil relación con la oscuridad que mejoraba la percepción del sonido y diluía las nociones de distancia y proximidad. Las dos puertas del contenedor estaban parcialmente abiertas; esto permitía una visión segmentada del horizonte y dejaba entrar un poco de luz y aire del exterior. Esta disposición ofreció un diálogo metafórico entre el interior y el exterior, que emulaba la superficie y las profundidades del océano.

El trabajo estuvo abierto en dos momentos. En la primera ocasión estuvo dispuesto sólo como un adelanto para un público de críticos de arte, y un mes más tarde fue abierto al público. Entre estas dos exhibiciones, técnicamente, la pieza redujo la cantidad de computadoras usadas para la ejecución; y, composicionalmente, se alteró muy poco apenas afinando algunos detalles sobre el equilibrio entre el sonido digital y la capa mecánica.

Exposición para grupo privado

El grupo evaluador llegó al lugar por la noche según lo solicitado para la experiencia. Sin ningún preámbulo o introducción experimentó la pieza, asimilándola sin ninguna dirección o guía durante una hora. El grupo pasó tiempo dentro del contenedor de carga, pero también caminó alrededor de él, escuchando la fusión de diferentes elementos acústicos desde el exterior y desde diferentes lugares, sobre todo, entre 10 y 30 metros de distancia.

Una vez que los asistentes terminaron de experimentar la obra se generó un espacio para el diálogo y la discusión. Primero escucharon mi explicación en la que describí la naturaleza de la obra y destacué varios puntos acerca de la composición, su organización y la toma de decisiones. Más tarde compartieron opiniones entre ellos, señalando observaciones, ideas y referencias. Al final, la comisión hizo algunas preguntas sobre mi trabajo y realizó algunas observaciones, culminando en una conversación sobre el arte sonoro y la instalación sonora.

Exposición pública

Un mes después de la presentación privada al comité, la pieza estuvo abierta al público durante tres días. Con cerca de 50 visitantes por día, la pieza generó valiosas interacciones con los visitantes porque la gente permanecía 20 minutos en promedio, escuchando el trabajo

dentro del contenedor y sintiendo las vibraciones del lugar. Algunos miembros de la audiencia caminaron alrededor del contenedor, incluso una persona decidió experimentar el trabajo desde la parte superior del contenedor y subió al techo.

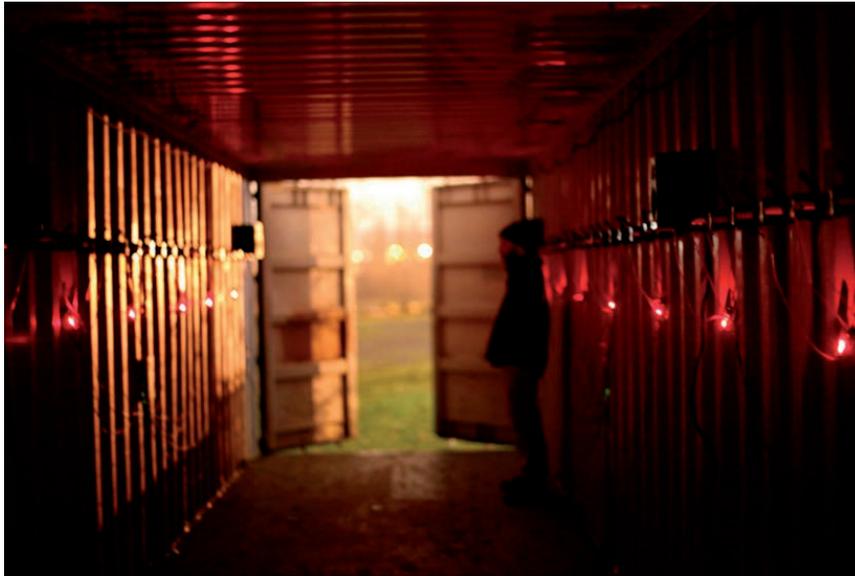


Figura 12.37. | Vista interna de la instalación A.

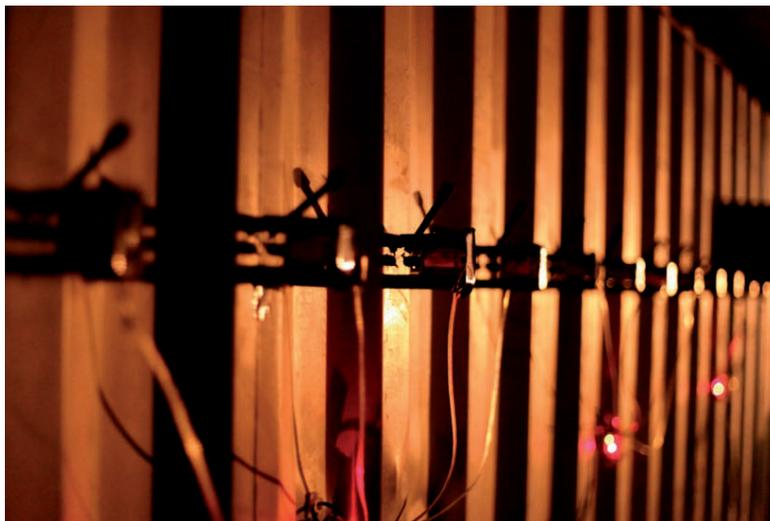


Figura 12.38. | Vista interna de la instalación B.

Plática de artista

La inauguración de la exposición pública fue complementada con una charla en la que expuse una visión general del proyecto, incluyendo el trabajo y la investigación realizada en el barco, el proceso de registro de los ventilas hidrotermales, el desarrollo de la tecnología de los módulos plurifocales y, por último, la idea de contar con el contenedor de carga en tierra. Unas 50 personas asistieron a la charla en el Laurel Village Center, ubicado justo al otro lado de la calle del contenedor de carga.



Figura 12.39. | Postal de invitación (frente).

Comentarios y retroalimentación

Muchos de los visitantes pertenecían a la comunidad oceanográfica y, algunos de ellos expresaron la sensación de que la instalación evocaba el mar. Aunque no era mi intención obtener una representación literal me resultó interesante y provechoso contar con tal inter-

AXIAL

A Sound Installation
by Hugo Solis

December 8th, 9th & 10th, 2011
Open from 4:18 PM to 11:00 PM

Axial is a sound installation where a cargo container is employed as a resonant object. Resonant in its acoustic sense because the container is used as sound generator but also resonant in its metaphorical connotation because the container and the generated sounds translate and represent the geological properties of Axial, an active submarine volcano on the Juan de Fuca Ridge located about 250 miles off the coast of Oregon. The container is also the space/instrument for interpreting the scientific data obtained during the oceanographic expedition Enlighten'10 where the artist recorded the sounds of the hydrothermal vents located in the area at depth of over 4500 feet.

Center for Urban Horticulture | UW Botanic Gardens | 3501 NE 41st St., Seattle, 98195
More info at <http://hugosolis.net>

DARTS

UNIVERSITY of
WASHINGTON

40
YEARS

Seattle Office of
ARTS &
CULTURAL AFFAIRS

4
CULTURE

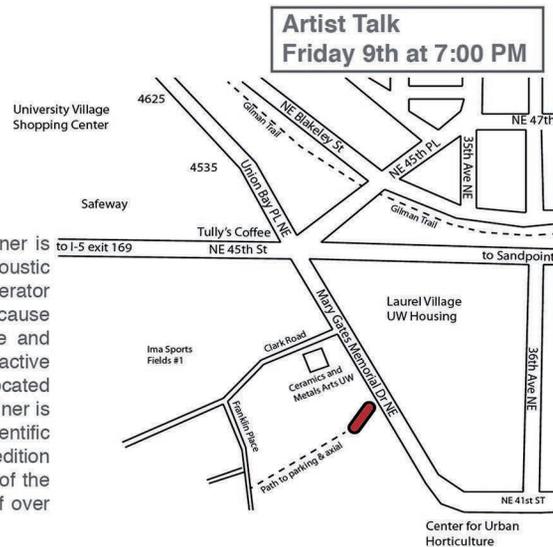


Figura 12.40. Postal de invitación (reverso).

pretación. Uno de los miembros del comité mencionó que el trabajo se parece al trabajo escultórico de Richard Serra (pionero de arte mecánico). Trimpin también visitó el trabajo y mencionó que le era inspirador y acústicamente interesante; además, habló conmigo acerca de la percusión controlada por dispositivos digitales, detallando cómo obtener paletas tímbricas variadas.

Conclusiones

No hay duda de que las principales razones para crear *Axial* fueron la curiosidad respecto al sonido y el deseo de explorar su potencial. El océano como detonador de imágenes, metáforas y fantasía impregnó el concepto general de la obra y dirigió la investigación y el desarrollo de la pieza.

Los bocetos y los experimentos iniciales comenzaron en 2008, lo que significa que me llevó más de cuatro años realizar el trabajo. Durante este proceso aprendí sobre el océano y sobre el fondo marino, estudié y analicé trabajos con conceptos similares previos y desarrollé la tecnología para llevarlo a cabo (tanto en hardware como en software) planificando la logística, creando obras secundarias donde los componentes pudieran ser probados individualmente y, finalmente, presenté y documenté la obra. El desarrollo de esta obra fue un proceso profundo y, en cierto sentido, un recorrido agotador acompañado de errores y revaloración de ideas.

Las aproximaciones sucesivas, las evaluaciones heurísticas, las negociaciones entre imaginación y realidad, las pruebas y los errores, así como la repetición de los procesos de trabajo fueron métodos para producir un resultado gradual y un trabajo procesado. Juzgar o evaluar el trabajo podría estar fuera del alcance de esta revisión, pero es evidente que *Axial* no es una obra simple o fugaz. Los desafíos, la infraestructura y el apoyo requeridos hicieron de *Axial* una propuesta que, desde lo más profundo, intenta contribuir a las perspectivas del arte sonoro.

Mientras que hay elementos dentro de la obra que se podrían haber logrado más óptimos y quizá hubiera podido llegar a ellos directamente, las aproximaciones sucesivas que le dieron forma a la pieza ayudaron a crear un espacio de experimentación que poco a poco se hizo más estrecho hasta obtener la versión final. Tal método, sin duda, no fue el medio ideal para producir un producto, pero ciertamente creó la mejor circunstancia para el aprendizaje y el descubrimiento. Pedagógicamente, la producción de este trabajo me dio un enorme conjunto de recursos y habilidades para la creación de futuras piezas.

Al igual que con cualquier obra de arte y cualquier propuesta estética este trabajo se deriva de un conjunto de trabajos anteriores y de propuestas creadas por otros artistas. El trabajo se esfuerza por combinar aspectos específicos de proyectos previos con el fin de generar una propuesta original bajo el entendimiento de que la originalidad proviene de la evolución y no de la generación espontánea. Escribir sobre el producto terminado me ha proporcionado la oportunidad de analizar la evolución del trabajo, y, también, me ha incentivado a señalar las buenas decisiones, así como aquellas menos afortunadas del proceso. Al igual que con cualquier esfuerzo humano este trabajo está sujeto a mejoras,

pero la asimilación que podemos adoptar ante el proyecto únicamente puede ser determinada mediante la perspectiva adquirida con el paso del tiempo.

La experiencia de crear *Axial* me aportó conocimientos en el campo de la tecnología para el arte y también devino un conjunto de mecanismos y procesos que los futuros artistas pueden aplicar a sus propios esfuerzos creativos.

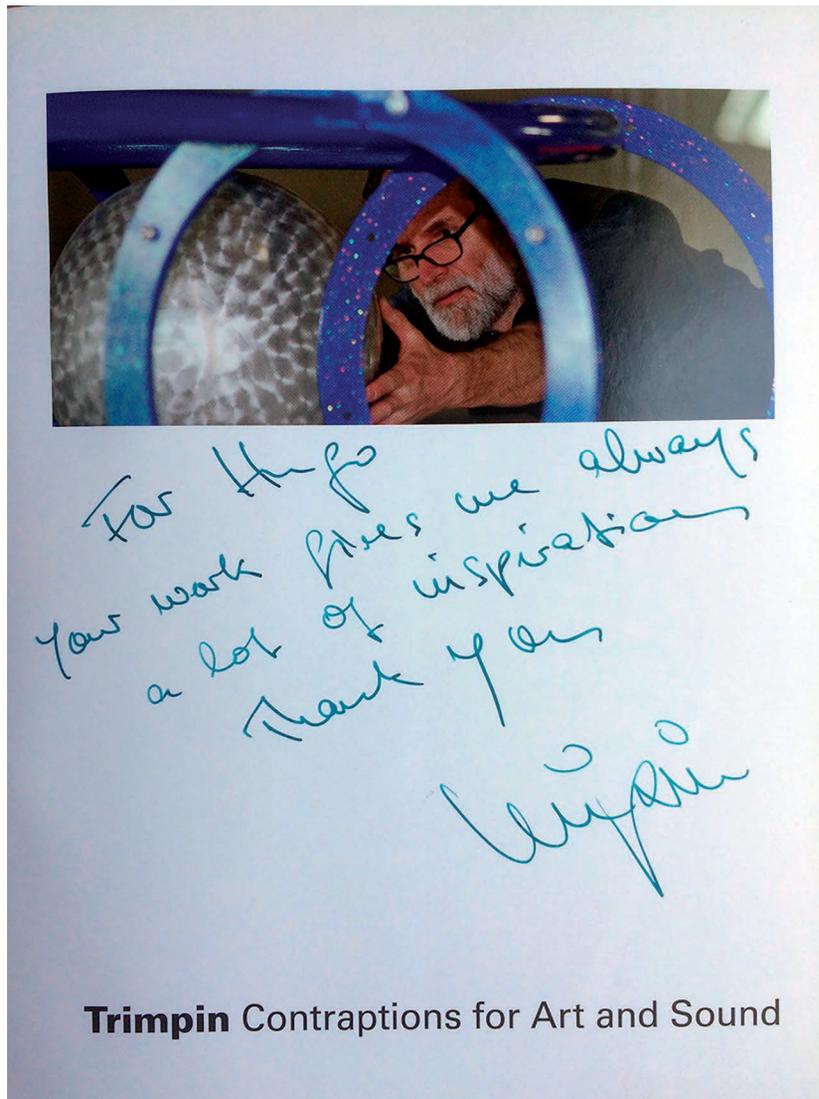


Figura 12.41. | Dedicatoria de Trimpin para el autor.

Contribuciones

Axial contribuye al campo del arte sonoro y el arte electrónico de varias maneras:

- La pieza como obra terminada, con su naturaleza interna, paradigma y concepto se suma a la lista de obras que exploran el sonido en relación con el espacio y los espacios públicos y amplía la lista de obras que tienen al océano como personaje principal detonador, creando un diálogo con la proporción humana y el entorno.
- El trabajo es un estudio de caso sobre el sonido y el espacio. Es un proyecto basado en obras y conceptos de varios artistas entre los que se incluyen: Pampin, Karpen y Barber. Este trabajo explora las lógicas y los matices del sonido y del espacio no sólo en el sentido acústico, sino también en el sentido conceptual y el social.
- El trabajo ofrece un conjunto de grabaciones de sonido de una ubicación en tierra que nunca se había grabado desde una perspectiva puramente artística. El trabajo de las grabaciones de las ventilas hidrotermales funcionan como un manifiesto de curiosidad y exploración.
- Una de las contribuciones técnicas de este trabajo consiste en la creación del Plurifocal Events Controller que ofrece una solución a la distribución de sonido analógico. Aunque hay varios medios para resolver esta situación específica, el PEC condensa el proceso porque fue creado con este objetivo preciso en mente.
- Otra contribución técnica consiste en la creación del software Juum, que facilita el control de tiempo de trabajos híbridos que requieren el uso de una línea de tiempo estructurada y extendida para controlar secuencias de comandos de código.
- Una tercera contribución técnica fue la creación de una infraestructura de software que aporta una solución de síntesis de partituras de audio. Esta solución (que utiliza el paradigma de la computación en la nube) se puede extender fácilmente si es necesario procesar una gran cantidad de datos. ■■■



Helix Aspersa

Hoja de sala

Fecha de producción: 2013. *Lugares de exhibición:* Fonoteca Nacional, CDMX; Museo Universitario de Arte Contemporáneo de la UNAM; Auditorio del Instituto Tecnológico y de Estudios Superiores de Monterrey. *Material:* Equipo de laboratorio fisiológico, micromanipulador, estirador de pipetas, amplificador de señales neurológicas, microscopio, ganglio cerebroide de caracol de jardín. *Tecnologías creativas:* Código para SuperCollider.

RESUMEN

En el proyecto *Helix Aspersa* el artista sonoro Hugo Solís y el neurofisiólogo Hugo Solís suman sus intereses y experiencia para proponer una obra en la que se sonifican los potenciales de acción de ocho neuronas de diferentes ganglios cerebroides, esto es, de ocho diferentes caracoles de jardín. Los patrones rítmico-sonoros son metafóricamente una representación acústica de nuestros impulsos neuronales que a su vez son los elementos básicos de nuestra propia conciencia y percepción del mundo como seres humanos.

La instalación consta de ocho ganglios aislados subesofágicos de *Helix Aspersa* (caracol de jardín) cuyas neuronas son registradas con el equipo electrofisiológico necesario. Cada módulo consta de una cámara de registro, un micromanipulador (dispositivo que desplaza

el electrodo en movimientos de micras), un electrodo-micropipeta, un amplificador de señales biológicas AC-DC, un canal de un osciloscopio, una bocina amplificada. Los ocho módulos se colocan en media luna sobre una mesa antivibratoria.

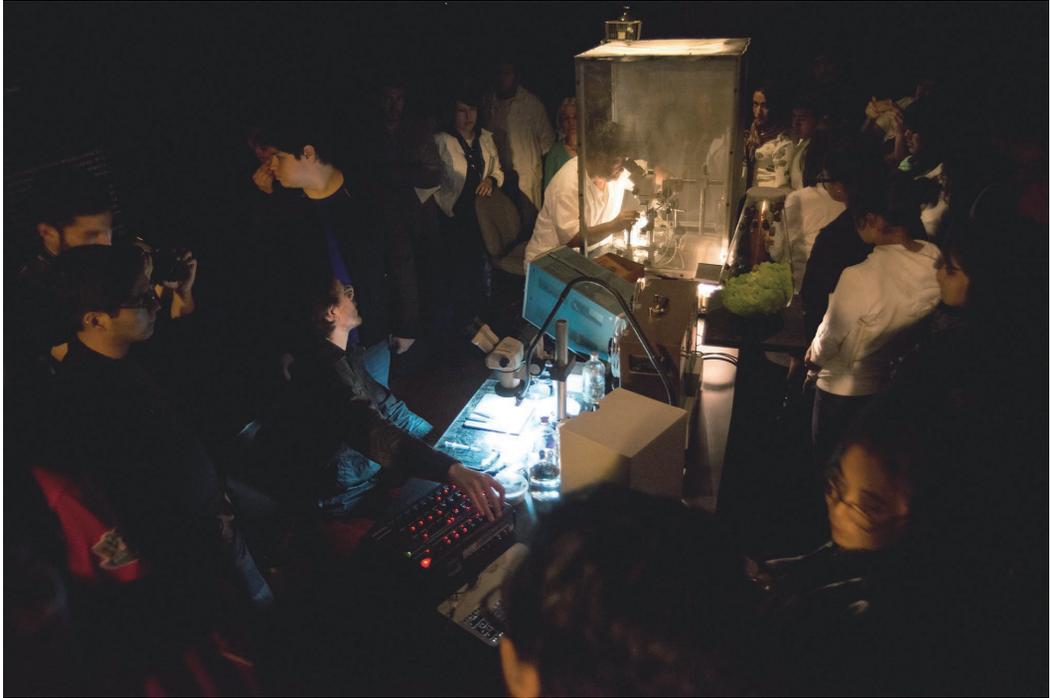


Figura 13.1. | Performance de *Helix Aspersa*.

El visitante entrará a una sala de exhibición semioscurecida en cuyo centro se encuentra una mesa de metal sobre la cual se halla un equipo de laboratorio neurofisiológico. Se observan ocho módulos similares y en cada uno de ellos vemos la concha de un caracol de jardín. Junto a la concha se observan una cámara de registro y un electrodo que conduce a un amplificador de señales y éste a una bocina amplificada. Acústicamente el visitante se encuentra con una experiencia polirrítmica que asemeja lluvia o granizo.

CONCEPTO

La neurona es la unidad fundamental del sistema nervioso central. Sus propiedades intrínsecas determinan el funcionamiento básico de la actividad neuronal. Esta actividad también se refleja en el comportamiento de una determinada red neuronal con propiedades dinámicas, expresivas e inteligentes. Por ejemplo, en ellas radica la habilidad que tienen los sistemas biológicos de mostrar aprendizaje, memoria como reflejo de cambios plásticos de corta y larga duración.

Las neuronas establecen comunicación entre sí debido a los contactos sinápticos en donde el potencial de acción libera un posible neurotransmisor que actúa sobre la membrana posináptica. Esta comunicación también opera bajo el principio de equilibrio entre la excitación y la inhibición. Cuando se rompe este equilibrio aparece el caos. De aquí la importancia de conocer el comportamiento bioeléctrico de una neurona aislada y caracterizar ampliamente sus propiedades intrínsecas, entre ellas su patrón de frecuencia de disparo, el umbral de excitabilidad, el fenotipo bioquímico y su morfología.

Estas propiedades se valoran mediante el registro de la actividad eléctrica a través de un microelectrodo colocado en el medio intracelular. Esta actividad se amplifica y se visualiza en un osciloscopio y, dado su rango de frecuencia y amplitud de potencial de acción, se puede hacer audible.

En el campo de las neurociencias este conocimiento se ha obtenido tanto en invertebrados como en vertebrados. Dentro de los invertebrados se ha utilizado con mucha frecuencia al caracol de jardín (*Helix Aspersa*) debido a que su sistema nervioso central está conformado por neuronas que alcanzan tamaños del orden de las 300 micras, lo que facilita la inserción del microelectrodo y su manipulación separadas del resto del animal (preparación *in vitro*).

No es fácil extrapolar pero resulta de gran valor considerar el hecho de que las neuronas del caracol exhiben un comportamiento similar a las neuronas de animales *superiores* como el ser humano.

Por lo anterior, el hecho de escuchar los patrones de los ritmos de las neuronas del caracol es metafóricamente la escucha de un sistema similar al que integra nuestro sistema neural el cual constituye nuestro mecanismo de percepción. Dicho en otras palabras, los ritmos y soni-

dos que emanan de esta instalación sonora son los mismos que nos permiten integrar lo que conocemos como percepción de la realidad, las emociones y el fenómeno estético. ■■■



Figura 13.2. | Nota de prensa en relación al proyecto.



Delta 57.00 / -99.00

Hoja de sala

Fecha de producción: 2014. *Lugares de exhibición:* Permanente en Playa Ventura, municipio de Copala, Guerrero México. *Material:* Cemento. Tule seco, madera de palma, microcomputadoras, sensores, bocinas y sistema electrónico. *Tecnologías creativas:* Código para Pure Data, Arduino y Raspberry.

Este proyecto fue realizado con el apoyo del “Programa de Fomento a Proyectos y Coinversiones Culturales 2013” del Centro de Cultura Digital de Conaculta y del Instituto Tecnológico y de Estudios Superiores de Monterrey con un proyecto NOVUS.

RESUMEN

Delta 16.57 / -99.00¹ es una instalación sonora ubicada en Playa Ventura, en el estado de Guerrero. Se trata de un escuchador análogo a los tradicionales miradores que invita a realizar una mirada acústica del mar. El objetivo principal de la obra consiste en la contemplación sonora de un entorno natural con un fondo acústico que, de primera instancia, pare-

¹ Este capítulo aparece íntegro en el libro *Comunidades y contextos en las teorías y prácticas artísticas contemporánea* (Benítez, 2015).

ce simple pero que gracias a una escucha atenta y dirigida ofrece una composición rica en eventos y narrativa.

A muy pocos kilómetros del lugar donde se encuentra la instalación, en las coordenadas 16.57 / -99.00 desemboca en el mar el río Copala. Allí, una barra espera cada año la época de lluvias para abrirse y liberar el agua dulce que el río ha acumulado en la laguna por largo tiempo. La imagen poética de un enamoramiento lento y constante que se colma cada año entre el agua salada y el agua dulce es la imagen detonadora de la narrativa y del concepto de la obra.

El ritmo de la naturaleza, los cambios meteorológicos del entorno natural –geológicos y estacionales– están más allá de la percepción humana directa y por ello utilizamos métodos alternos para su observación. En este caso, los sonidos electrónicos son un contrapunto al presente generados con materiales meteorológicos que incluyen datos del momento actual, del tiempo pasado y de inferencias del futuro.

Físicamente la obra consta de una estructura elevada de forma parabólica la cual favorece que el sonido puro del mar se amplifique de manera natural. Simultáneamente, el sonido natural se combina con sonidos generados de forma digital. Estos sonidos digitales son derivados de la información meteorológica que reciben los sensores y que perciben humedad, cantidad de lluvia, color del mar, presión, dirección y velocidad del viento, luminosidad y temperatura. Al tiempo que se recaba la información meteorológica antes descrita, una serie de microcomputadoras analiza la información y genera los sonidos electrónicos.

Delta 16.57 / -99.00 es por sí sola una obra sonora; sin embargo, gesta un proyecto más amplio llamado *Sonoridad Delta* que invita a la apropiación del espacio para realizar obras de arte ecológico-sustentable en este sitio haciendo del espacio un lugar público. El espacio está abierto a la comunidad de artistas que requieran el entorno para realizar sus propuestas artísticas.

Este proyecto fue realizado con el apoyo del Programa de Fomento a Proyectos y Coinversiones Culturales 2013 del FONCA, del Centro de Cultura Digital de Conaculta, y del Instituto Tecnológico y de Estudios Superiores de Monterrey. Más información y detalles se pueden consultar en <http://www.SonoridadDelta.org>.

INTRODUCCIÓN

Desde los años sesenta del siglo xx ha ido en aumento la proliferación de experiencias sonoras, que se han ido clasificando bajo el concepto de arte sonoro. Bajo esta definición se han catalogado trabajos que tienen un fundamento y una base acústica, pero que suelen estar complementadas y en diálogo constante con otros medios expresivos, y, sobre todo, que trabajan el concepto de temporalidad de manera no lineal, ya que esta característica los diferencia de las experiencias consideradas puramente musicales.

De la misma manera los trabajos artísticos donde se vinculan conceptos relacionados con la ciencia y con la creación de tecnología, cuya finalidad es detonar experiencias creativas, ha generado, también en los últimos 50 años, la proliferación de expresiones que se agrupan dentro del concepto arte-ciencia-tecnología.

Por último, bajo la premisa de que cualquier manifestación de conocimiento o actividad humana es susceptible de convertirse en una experiencia expresiva si se enmarca adecuadamente en un contexto específico, utilizando metáforas y las conexiones conceptuales adecuadas, han surgido experiencias y definiciones por demás atractivas. Una de ellas, por ejemplo, es el denominado *land-art* (traducido como arte terrestre) cuya función trata de definir proyectos realizados en exteriores con materiales de la naturaleza, buscando generar un diálogo entre el entorno, la naturaleza y el espectador activo. En este espacio describimos una obra realizada durante 2014, que conjunta los tres campos descritos anteriormente: arte sonoro, arte-ciencia-tecnología y *land art*. *Delta 16.57 / -99.00* es una obra que tiene componentes de arte sonoro porque está concebida como una instalación sonora; tiene componentes de arte-ciencia-tecnología porque requiere un desarrollo tecnológico particular, así como el uso y la manipulación de información meteorológica con fines puramente artísticos; y, por último, tiene componentes del campo del arte terrestre porque busca, en cierta medida, generar un diálogo muy particular con el espacio que la rodea, además de que la obra en sí misma nació del entorno natural donde está instalada; es, en este sentido una pieza para sitio específico.

El documento presenta los antecedentes que detonaron la producción de la obra; el proceso creativo que dio cauce a la propuesta final; el proceso de construcción que requirió una toma

de decisiones prácticas; el desarrollo tecnológico que describe el funcionamiento de la parte electrónica; el trabajo de composición y sonificación que describe la toma de decisiones y el desarrollo en el trabajo de traducción entre los datos y la generación del material sonoro digital; el trabajo comunitario que se realiza en la comunidad y que acompaña la obra artística; y, finalmente, una serie de conclusiones y trabajo futuro por realizar.

ANTECEDENTES

Entre 2006 y 2011 el autor vivió en la ciudad de Seattle mientras realizaba la parte presencial de un doctorado en Arte Experimental y Medios Digitales. En 2008 se definió la línea de investigación que se concentró en la creación de una propuesta sonora que utilizaba datos oceanográficos y grabación de ventilas hidrotermales localizadas 1 500 metros bajo el nivel del mar. Para dicho proyecto fue necesaria una estrecha colaboración y la participación de un equipo de oceanógrafos miembros de la Escuela de Oceanografía de la Universidad de Washington, trabajo que en el capítulo 12 se describe a detalle.

Una de los elementos de dicho trabajo consistió, como ya se mencionó, en la sonificación de datos que los oceanógrafos obtuvieron durante la expedición al volcán submarino llamado Axial localizado a 400 kilómetros mar adentro, a la altura de Portland, Oregon. Dicha información no es de naturaleza sonora y su origen y su interpretación no suelen derivar en un material auditivo. Sin embargo, siendo valores en su mayoría continuos que, a su vez, cambian a lo largo del tiempo, son candidatos para ser utilizados como moduladores de parámetros sonoros.

De esta manera fue que el autor inició una exploración de larga duración que involucra el uso de información generada por entornos naturales como detonadores creativos cuya función opera como organizadora de elementos sonoros. Esta metodología le sirve al autor como mecanismo para generar una intersección coherente entre las ciencias y las artes, al mismo tiempo que permite generar obras con formas y estructuras orgánicas donde, manteniendo cierto nivel de autoría, son generadas parcialmente por sí mismas. En este sentido, el autor es un catalizador y regulador de funciones preexistentes en el entorno más que un organizador de materiales seleccionados de manera arbitraria o azarosa.

Al regresar a México el autor continuó atraído por dicha estética y por la aplicación de las tecnologías antes mencionadas. Al mismo tiempo seguía atraído por la seducción y el potencial poético del océano y, por tanto, deseaba continuar con labores en dicho entorno. En abril de 2013 realizó una visita vacacional a Playa Ventura, lugar que solía frecuentar durante los años 90 y en el cual existía un terreno abandonado propiedad de la familia. Durante dicha visita se vislumbró la posibilidad de aprovechar el lugar y generar una propuesta artística que le diera continuidad al trabajo realizado en Seattle desde una perspectiva diferente. Si el proyecto *Axial* se centraba en la profundidad del océano, este nuevo proyecto tal vez debería centrarse en la superficie y en la costa. En este sentido, se cumpliría la intención de “tropicalizar” los conceptos previos y al mismo tiempo se generaría una obra dentro de la cosmogonía del agua y del mar como devenir directo de *Axial*.

La idea se mantuvo como una tentativa y como propuesta durante varios meses, pero no fue sino hasta principios de 2014 que finalmente se contó con los recursos para llevarla a cabo a través de una propuesta dentro del Programa de Fomento a Proyectos y Coinversiones Culturales del Fondo Nacional para la Cultura y las Artes 2013.

PROCESO CREATIVO

La propuesta inicial presentaba un boceto que, en líneas muy generales, proponía una instalación sonora que debía montarse justo enfrente del mar y que debía generar un diálogo entre el océano, los espectadores y el entorno de la naturaleza. Además, el espacio debía funcionar como un lugar de reunión que permitiera a la comunidad de artistas electrónicos experimentar en un entorno con dichas propiedades.

El hecho de que en un lugar muy cercano al poblado desembocara el río Copala en el mar y que, adicionalmente, fuera cuna de una comunión marítima es una seducción en sí misma: un delta cercano plétórico de vida. Dichos lugares se conocen como deltas y son lugares con propiedades muy particulares porque se forman justo en el vértice donde el agua dulce se junta con el agua salada. Suelen ser espacios de mucha actividad productiva y atraen con frecuencia a pescadores dada la concentración de vida. El hecho de saber que

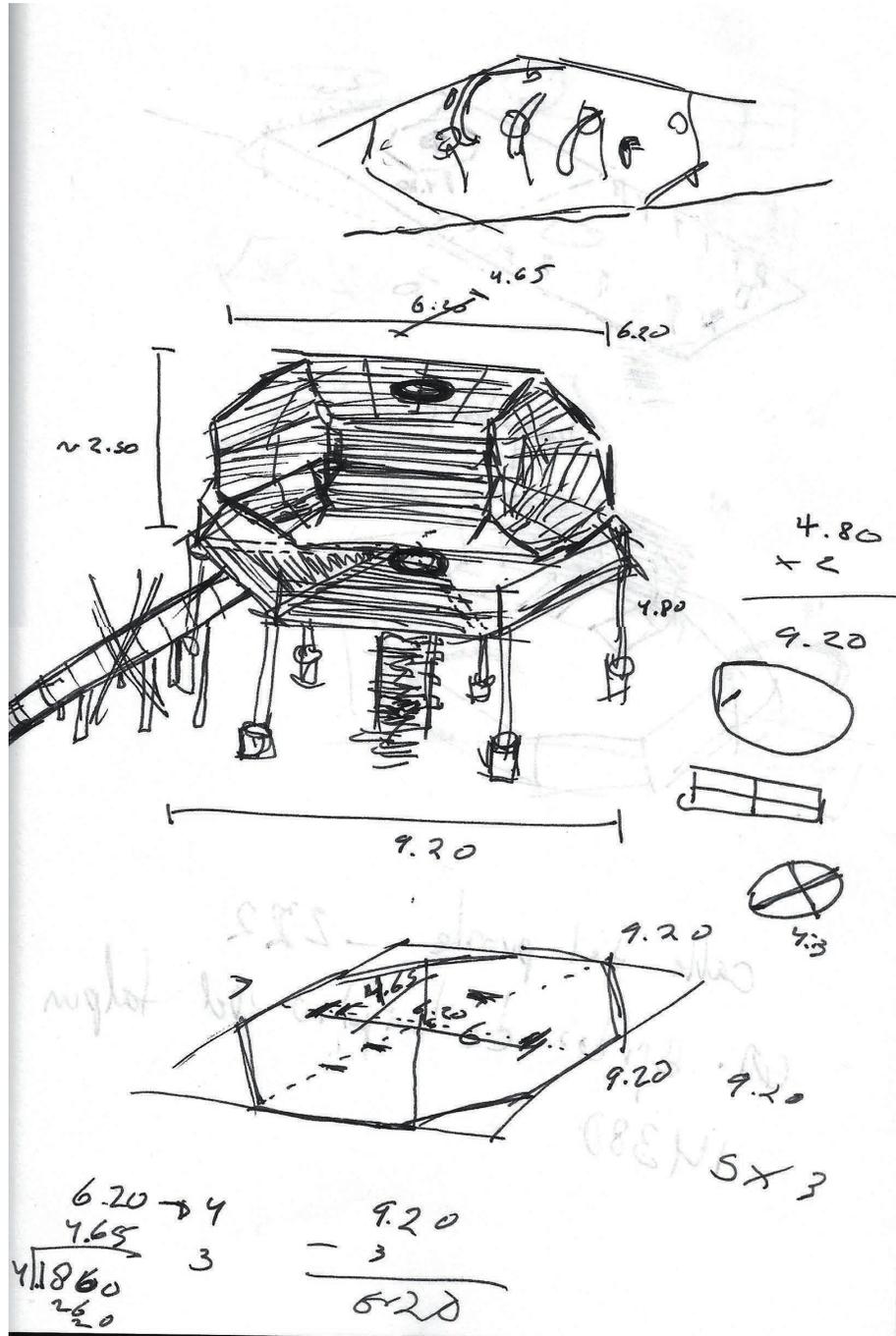


Figura 14.1. | Boceto del proyecto Delta.

había un delta en la zona le dio nombre al proyecto, y, poco a poco, definió parte del contenido poético del mismo.

En esa etapa del proceso era muy importante realizar un proceso creativo minucioso y reflexivo con el fin de clarificar los detalles y las precisiones requeridas antes de iniciar la construcción de la pieza. En enero de 2014 se realizó una primera visita de reconocimiento en la cual se limpió y preparó el espacio, al tiempo que se realizó un análisis del entorno y sus propiedades.

De enero a principios de mayo de 2014 se realizó un curso en conjunto con dos alumnos del Tecnológico de Monterrey en la modalidad de Estancias de Investigación en el cual se abordaron obras relacionadas con el arte y el entorno. Durante esta fase se revisaron piezas icónicas del *Land-art* y se realizaron diversos bocetos y propuestas con una serie de premisas, entre las que se encontraban que el sonido jugaría un papel preponderante y que la obra debía de producirse con la mayor cantidad de material de la región. Se sumaba el detonante de que la pieza debía generar un diálogo con el entorno y, en la medida de lo posible, debía vincularse con el delta ubicado a siete kilómetros del espacio del montaje. La pieza debía de ser, al mismo tiempo, una instalación sonora y funcionar como nodo o espacio de trabajo para futuros proyectos comunitarios.

Durante dichos meses se trabajaron muchas alternativas, entre las más relevantes por mencionar se encuentran aquellas que consideraban componentes electrónicos, algunas híbridas y otras, las menos, analógicas en su totalidad. Uno de los temores fundamentales por considerar fue la dificultad de mantenimiento en un entorno extremadamente húmedo y con erosión y oxidación constantes provocada por la brisa de agua salada. Se contemplaron propuestas basadas en energía solar y otras en energía eólica. En relación al uso de datos se vislumbró el uso de sensores meteorológicos para la obtención de información susceptible de manipulación y sonificación. Los bocetos y las propuestas que se fueron descartando durante el proceso se pueden revisar en el portal web del proyecto.

Para la primera semana de mayo se contaba ya con un concepto claro y un boceto de cómo debería ser la obra. El concepto de hacer un “escuchado” análogo a los miradores que existen como puntos para la contemplación adquirió fuerza y, con ello, la forma y la funcionalidad del espacio. Así inicia la etapa de construcción de *Delta*.

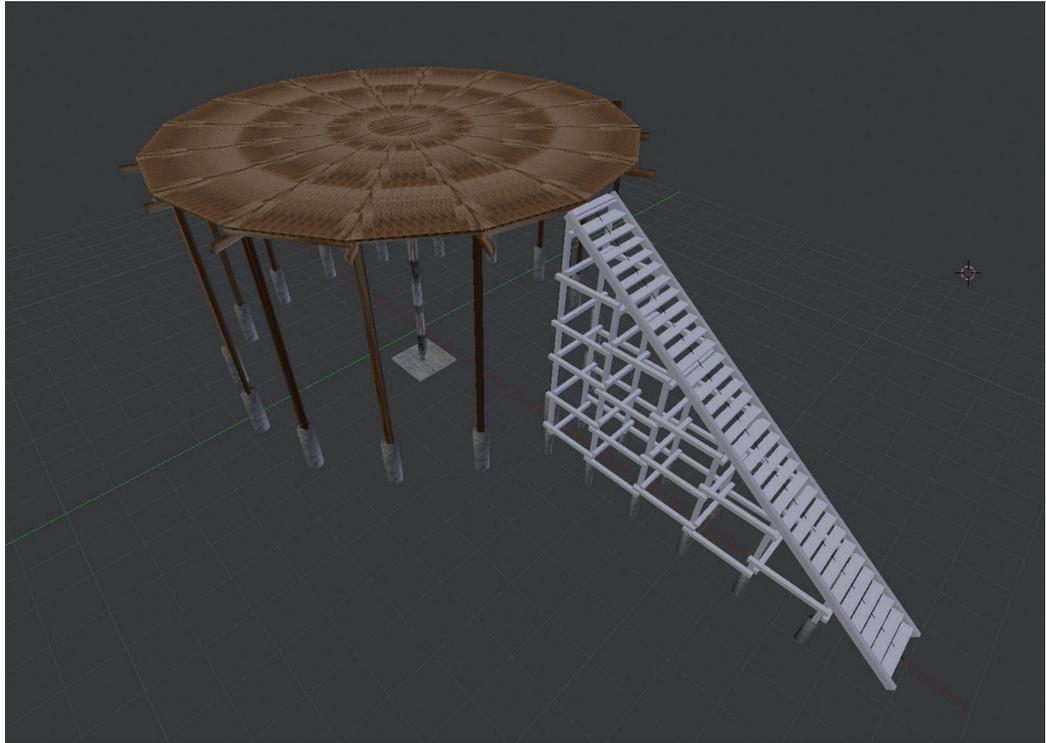


Figura 14.2. | Render en Blender del proyecto *Delta*.

DESARROLLO DE CONSTRUCCIÓN

Para la propuesta establecida fue necesario crear una plataforma de varios metros de altura. Se descartaron el fierro y el cemento por el difícil mantenimiento que implican debido a su ubicación frente al mar. Inicialmente se pensó en utilizar madera tratada, pero al poco tiempo fue evidente que lo mejor y más natural era utilizar madera de palma de la región. Esta madera apoyaba parte del propósito sustentable de la obra, ya que los campos de palma deben renovarse periódicamente y, al formar parte de este ciclo, la vida de la pieza se unía al ciclo natural del entorno. Al poco tiempo se estableció contacto con una familia que se dedicaba al corte de madera de palma y se estableció un trato en el cual dicha familia proveería la madera a cambio de que el grupo de trabajo pudiera vivir el proceso de corte de la madera de manera directa. Se cuenta con registros y con documentación en video de dicho proceso que se pueden ver en el portal web de la obra.

La construcción de la base estuvo a cargo de un maestro del asentamiento El Papayo, localizado a tres kilómetros de Playa Ventura. Guillermo y sus ayudantes realizaron el proyecto establecido basados en unos planos tridimensionales creados por el artista y validados por un grupo de arquitectos. Inicialmente la intención era crear una base totalmente de palma, pero al terminar el trabajo se evidenció que la estructura no era lo suficientemente estable y, por tanto, la palmera central tuvo que ser recubierta de cemento. Es importante destacar que dentro del corazón del cimientto está la palmera que le da la fortaleza poética y la estructura medular que funciona como vaso comunicante con la tierra. Al terminar la base se diseñó la escalera y un barandal muy mínimo por cuestiones de seguridad. Toda la estructura fue cubierta con barniz epóxico y se inició el proceso de construcción de la estructura principal.



Figura 14.3. | Proceso de construcción de la estructura de *Delta*.

En visitas a museos de ciencias el autor había vivenciado el efecto de amplificación de audio al estar dentro de estructuras parabólicas. Ante esa experiencia se optó por una estructura que en su interior concentrara el sonido en un punto focal y que con ello amplifi-

cara el sonido del mar. Hacer una estructura robusta en forma parabólica tiene ciertas dificultades de construcción, en especial si se desea realizar con medios sustentables.



Figura 14.4. | Construcción de estructura del escuchador.

La idea original era realizarla con desechos de botellas PET a fin de ayudar en la recolección de basura. Esta idea estaba basada en las construcciones realizadas por el equipo de trabajo de Germinalia A.C., idea que fue descartada para sustituirla por la idea de hacerla con paja repellada de forma análoga a la construcción que se encuentra a dos kilómetros de la obra. También se consideró realizar el relleno con la hoja de palma con la cual se hacen los techos de las palapas de la región.

Finalmente, a sugerencia de un habitante de la región, se utilizó hoja de tule para darle cuerpo a una estructura que se hizo con fajillas de pino en la Ciudad de México. Es importante mencionar que, paralela a la carretera, a un costado del poblado y colindante con el



Figura 14.5. | Secado de tule obtenido de la laguna.

terreno, hay una gran laguna que se irriga con el río Copala. Hace algunos años esta laguna se encontraba saludable: estaba abierta y se podía pescar en toda su superficie. Sin embargo, poco a poco, la planta de tule ha proliferado a manera de plaga y ha invadido una extensa área, limitando las zonas de acceso. Aunque ha habido algunas iniciativas por parte de la comunidad y del gobierno para remover el tule y sanear la laguna, generando pastura como alimento para animales; el proceso es extremadamente costoso. Mientras tanto, los habitantes han empezado a secar la hoja de tule y han sustituido la hoja de palma por esta hoja. Parece ser que el tule es más resistente y se evita la proliferación de alacranes. Con este conocimiento se optó por cortar tule, secarlo al sol y utilizarlo como interior de la estructura.

Una vez terminada la parte interna de la estructura fue repellada con cemento y posteriormente pintada. El efecto acústico funcionó conforme a lo planeado y existe un punto específico donde el espectador puede escuchar un efecto de desplazamiento y movimiento del sonido. El viento y el oleaje, además de ser amplificados, pasan por una serie de resona-



Figura 14.6. | Repellado de estructura del escuchador.

dores naturales y de generadores de movimiento en cuanto a la percepción de la fuente sonora se refiere. En esta etapa de la obra la parte analógica se concluye y ya se cuenta con el espacio para la exploración digital.

DESARROLLO TECNOLÓGICO

Una vez terminada la estructura física y habiendo comprobado el efecto de amplificación y resonancia, se diseñó la estructura electrónica. Se contempló el uso de bocinas hiperdireccionales a base de transductores de ultrasonido, pero este proceso se descartó al poco tiempo debido a su costo y a su limitado rango de frecuencias. En 2012 el autor había empezado a experimentar con transductores que permiten la vibración de estructuras sólidas como ventanas y mesas. En 2013 se estableció contacto con la empresa mexicana

Alfacústica la cual se dedica al montaje de bocinas invisibles. Aunque inicialmente se pensó en empotrar y ocultar los paneles de vibración se decidió dejarlos expuestos, favoreciendo la calidad de sonido y la simplicidad del mantenimiento. Los transductores fueron montados en estructuras de aluminio a los paneles de vibración y éstos montados en marcos y empotrados en la parte interna de la estructura parabólica. Se montaron cuatro paneles, dos a los lados, uno en la parte trasera y uno en la parte superior de la zona de mejor escucha dentro de la concha acústica.



Figura 14.7. | Pruebas con amplificadores para las bocinas.

Se diseñó también una pequeña torre de medición meteorológica capaz de registrar en tiempo real los valores de temperatura, humedad relativa, presión, luminosidad, velocidad del viento, dirección del viento y precipitación pluvial. Este sistema se basa en un circuito comercial que se alimenta por una pila de litio que, a su vez, se mantiene cargada a través de un panel solar. Los datos son transmitidos vía Zigbee usando un Arduino y un Xbee. Ini-

cialmente se contempló colocar dicha torre justo en el delta del río Copala a siete kilómetros del escuchador, pero no existían las condiciones de seguridad necesarias.



Figura 14.8. | Montaje de bocinas.

Por otra parte, se diseñó una central de análisis de datos y de generación de audio compuesta por tres microcomputadoras Raspberry, dos amplificadores de audio estéreo-fónicos, tarjetas de sonido USB, una cámara HD, un micrófono de cuatro canales, fuentes de poder y un receptor Zigbee conectado a un Arduino. Una de las microcomputadoras se encarga de recibir vía Zigbee, a través del Arduino, los datos meteorológicos enviados desde la estación meteorológica colocada en la misma base a un lado de la concha acústica. Una vez recibidos, los datos meteorológicos son analizados, filtrados y enviados por medio de una red interna a las otras dos microcomputadoras. La misma microcomputadora se encarga de tomar fotografías y de mantener conexión con un servidor web vía internet. Inicialmente se consideró el uso de un módulo inalámbrico de conexión a la web 4G pero finalmente se pudo contar con internet vía telefónica.

Las otras dos microcomputadoras tienen cargado un sistema operativo optimizado para el procesamiento y síntesis de audio. Cada una de las computadoras cuenta con una tarjeta



Figura 14.9. | Sistema electrónico.

USB de audio externa para mejor calidad de grabación y reproducción, y ejecuta un Patch en el software de síntesis de audio PureData que reacciona y varía parámetros sonoros en función de los datos meteorológicos recibidos y del color del mar registrado con la cámara. Estas microcomputadoras se encargan, simultáneamente, de grabar sonidos y realizar transformaciones digitales de dichas muestras. Cada una de las microcomputadoras reproduce y graba dos de los cuatro canales del sistema.

COMPOSICIÓN Y SONIFICACIÓN

Una de las partes más interesantes durante el proceso de trabajo fue la composición de la parte sonora digital. El escuchador cuenta con cuatro bocinas en su interior conectadas a un sistema de amplificación cuadrifónico. El sonido es generado por medio del programa PureData y los valores que se utilizan como moduladores de la señal son, como se dijo anteriormente, la temperatura, el color del mar, la humedad, la presión, la intensidad lumínica, la velocidad y la dirección del viento, y la precipitación pluvial.



Figura 14.10. | Montaje del sistema electrónico.

La intención de la obra desde su gestación fue crear un sutil diálogo entre el sonido amplificado del mar natural y la parte digital, y por ello se descartó la producción de texturas electrónicas disímiles o fuera de contexto que no tuvieran relación con la parte germinal de la obra. Se buscaba mantener muy evidente la relación directa con el sonido amplificado del mar y su sonoridad. Por tal motivo, se decidió trabajar con texturas de ruido blanco filtrado que tienen mucha similitud con el sonido del mar. Los valores de frecuencia central y de ancho de filtraje serían los parámetros a correlacionar con los valores meteorológicos.

Como primera versión de la obra se decidió crear un banco de generadores de ruido blanco donde cada uno se filtrara en función del valor de cada uno de los parámetros recibidos en tiempo real. Los parámetros de mayor velocidad de cambio se generan en las bocinas izquierda-derecha y los de menor velocidad en las bocinas superior-trasera. Los rangos fueron calibrados empíricamente y ajustados *in situ* después de escuchas recurrentes. En una

versión posterior se espera integrar una base de datos que permita la recuperación de valores pasados y, con ello, hacer empalmes de situaciones meteorológicas diferentes de las actuales. Para lograrlo se requiere someter la información a procesos de inferencia y, con ello, crear predicciones y jugar con estructuras sonoras asociadas a valores meteorológicos que se estiman posibles en un futuro cercano.

PROYECTO COMUNITARIO

Aunque la parte central del proyecto es la instalación sonora *Delta 16.57 / -99.00*, la obra se enmarca dentro de un proyecto más general llamado *Sonoridad Delta* que está compuesto de dos elementos: el factor social y el de cooperación colectiva. Por un lado, el terreno y el espacio son ofrecidos de manera gratuita (mediante reservación y solicitud previa) a la comunidad creativa para el desarrollo de proyectos que requieran o se beneficien de un entorno como en el que está situada la obra. El programa está dirigido a



Figura 14.11. | Taller de electrónica en Playa Ventura.

personas que puedan aprovechar el espacio, trabajando en temas de arte y ecología, arte de la tierra, arte y entorno. Para ser beneficiarios del proyecto deben enviar su propuesta y su calendario de trabajo al portal www.SonoridadDelta.net.

Por otra parte y como un trabajo social de beneficio colectivo, el proyecto está acompañando de talleres de electrónica para la comunidad de Playa Ventura. El primer taller fue realizado en abril del presente año. Inicialmente fue planeado para jóvenes, sin embargo, al no establecer requisitos, los asistentes que se inscribieron fueron niños de entre seis y 12 años. La razón de esto obedece a que los jóvenes de la región trabajan desde temprana edad ayudando a sus padres en oficios y actividades cotidianas de las que les es prácticamente imposible deslindarse. Aunque la dinámica del taller derivó en algo distinto de la propuesta inicial, el taller fue de gran valor social y, a través de ejercicios y juegos, los niños aprendieron conceptos básicos de electrónica e hicieron prácticas en el uso del cautín y del multímetro. Algunos resultados de dicho taller se pueden consultar en el portal antes mencionado. Este primer taller fue realizado en la Casa de Cultura de Playa Ventura con el apoyo de la Secretaría de Turismo del municipio de Copala. Se contempla la realización de talleres posteriores, pero en esta ocasión en el espacio donde se halla el escuchador, con el propósito de sensibilizar a la comunidad en cuanto a la razón y naturaleza de la obra instalada.

Finalmente, detrás de los talleres se encuentra un deseo a largo plazo de generar un trabajo que denominamos *Tecnología rural* el cual busca ofrecer a la comunidad las bases de conocimiento para que quienes la habitan puedan apropiarse de las herramientas digitales necesarias para proponer, desde el interior de su organización comunal, soluciones a problemas reales sin una intervención paternalista que pudiera llegar a desconocer la naturaleza de los problemas locales.

CONCLUSIONES Y TRABAJO FUTURO

El proyecto *Delta 16.57 / -99.00* fue inaugurado el 2 de abril de 2015. Después de un año de desarrollo y dos años desde las primeras ideas para gestarlo, la obra fue terminada y puede ser apreciada por cualquier visitante. Se espera que se convierta en un atractivo y en un espacio de contemplación para la comunidad.

Durante las diferentes etapas del desarrollo de la obra a lo largo de 2014 se vivió un trabajo de constante creatividad y exploración en sitio. Algunas partes se realizaron en solitario y otras fueron un trabajo colaborativo en equipo. El equilibrio entre creatividad, fantasía e implementación práctica se dio de una manera muy natural y gradualmente se gestó lo que ahora es una obra terminada. Como en cualquier proceso creativo algunas decisiones pudieron ser diferentes u ofrecer resultados distintos. Sin embargo, la obra, tal y como fue finalizada, cuenta con elementos valiosos y una organización funcional que se sostiene en sí misma. La obra aborda una temática particular desde una óptica original e integra una serie de conceptos desde una perspectiva formal, pero, es de un componente poético, contemplativo y sutil.



Figura 14.12. | Documentación de la pieza *Delta*.

En este momento no se tiene certeza en cuanto al futuro de la obra. Se requiere un esquema de mantenimiento que, hasta este momento, no se ha contemplado. Se tiene la inquietud de continuar con los talleres y con el trabajo en el desarrollo de tecnología rural, y, sobre todo, se tiene el deseo de darle vida al espacio donde está la obra ofreciéndolo a la comunidad de artistas interesados en estos entornos.

Por otra parte, una de las actividades que pueden resultar interesantes de explorar en el futuro es generar modelos de sonificación más sofisticados que permitan empalmar condiciones meteorológicas, ya sea del pasado, ficticias o predichas con las condiciones reales. También se contempla invitar a otros artistas sonoros a proponer diferentes métodos de sonificación para contar con una colección de trabajos para esta obra en específico.

AGRADECIMIENTOS

Para la realización de este proyecto fue necesaria la participación y ayuda de un gran número de personas. Se desea agradecer al maestro Guillermo y familia, a Hugo Solís Ortiz, Isabel García, Gabriela Villa, Mateo Solís, Azucena, Indalecio, Celsa, René Cuevas, Romain Re y Claudia, Diego Ortega, Salvador Chávez, Enrique García, Guillermo Sánchez, Samuel Naranjo, Cecilia Sánchez, Ulises Calderón, Centro de Cultura Digital, Instituto Tecnológico y de Estudios Superiores de Monterrey, Ayuntamiento de Copala y habitantes de Playa Ventura. ■■■



Concepto: Hugo Solís García*
Realización: enero 2014-abril 2015
Material: madera de palma, tule, cemento, microcomputadoras, transductores y sensores climáticos.

Agradecimientos: Guillermo y familia, Gabriela Villa, Mateo Solís, Hugo Solís Ortiz, Isabel García, Azucena, Indalecio, Celsa, René Cuevas, Romain Re y Claudia, Diego Ortega, Salvador Chávez, Enrique García, Guillermo Sánchez, Samuel Naranjo, Cecilia Sánchez, Ulises Calderón, Centro de Cultura Digital, Instituto Tecnológico y de Estudios Superiores de Monterrey, Ayuntamiento de Copala y habitantes de Playa Ventura.



English information at
www.SonoridadDelta.net

DELTA 16.57 / -99.00

Delta 16.57 / -99.00 es una instalación sonora pública, se trata de un escuchador análogo a los tradicionales miradores que invita a realizar una mirada acústica del mar. El objetivo principal de la obra consiste en la contemplación sonora de un entorno natural con un fondo acústico que, de primera instancia parece simple, pero que, gracias a una escucha atenta y dirigida, ofrece una composición rica en eventos y narrativa.

A muy pocos kilómetros de aquí, en las coordenadas 16.57 / -99.00 desemboca el río Copala en el mar. Allí, una barra espera cada año, la época de lluvias para abrirse y liberar el agua dulce que el río ha acumulado en la laguna por largo tiempo. La imagen poética de un enamoramiento lento y constante que se colma cada año entre el agua salada y el agua dulce es la imagen detonadora de la narrativa y del concepto de la obra.

Los tiempos de la naturaleza, los cambios meteorológicos del entorno natural, geológicos y estacionales están más allá de la percepción humana directa, y, por ello, utilizamos métodos alternos para su observación. En este caso, los sonidos electrónicos son un contrapunto al presente, ya que se generan con materiales meteorológicos que incluyen datos del momento actual, del tiempo pasado y de inferencias del futuro.

La obra consta de una estructura de forma parabólica elevada donde el sonido puro del mar se amplifica de manera natural gracias a la forma de la estructura. Simultáneamente, el sonido natural se combina con sonidos generados de forma digital. Estos sonidos digitales son derivados de la información meteorológica que reciben los sensores de humedad, cantidad de lluvia, color del mar, presión, dirección y velocidad del viento, luminosidad y temperatura. Una microcomputadora analiza dicha información y genera los sonidos electrónicos.

Delta 16.57 / -99.00 es por sí sola una obra sonora; sin embargo, gesta un proyecto más amplio llamado *Sonoridad Delta* que invita a la apropiación del espacio para realizar obras de arte ecológico-sustentable en este sitio. El espacio está abierto a la comunidad de artistas que requieran del entorno para realizar sus propuestas (con previa solicitud y permiso). Para más información, dudas o comentarios, favor de visitar la página www.sonoridadDelta.org.

Este proyecto fue realizado con el apoyo del Programa de Fomento a Proyectos y Coinversiones Culturales 2013, del Centro de Cultura Digital de CONACULTA y del Instituto Tecnológico y de Estudios Superiores de Monterrey.

El acceso es libre y gratuito. Favor de no tirar basura y cuidar este espacio. La visita de niños pequeños deberá realizarse bajo una estricta supervisión de un adulto. Los miembros del proyecto no se hacen responsables de los accidentes que pudieran tener los visitantes ni de las consecuencias que pudiera tener el uso inadecuado del espacio. Favor de reportar cualquier desperfecto al correo electrónico info@sonoridadDelta.net

Proyecto apoyado por el Fondo Nacional para la Cultura y las Artes
**Miembro del Sistema Nacional de Creadores de Arte*

CONACULTA FONCA

Figura 14.13. Cartel del proyecto.

GAB

INTRODUCCIÓN

En 1997 el inventor de instrumentos musicales, Trimpin, realizó una serie de pláticas en la Escuela Nacional de Música dentro de las Jornadas Conlon Nancarrow. Estas pláticas estuvieron acompañadas de algunas demostraciones de un equipo electrónico capaz de tocar cualquier piano acústico por medio de pequeños solenoides que eran colocados sobre las teclas del piano y controlados por medio de una computadora. Este dispositivo había sido diseñado y construido en su totalidad por Trimpin quien es músico de formación, pero posee sólidos conocimientos en el área de la ingeniería, la electrónica y la computación. Su particular visión de la música y sus conocimientos en otras áreas le han permitido crear un lenguaje musical propio y original.

Las pláticas de Trimpin fueron un evento significativo en mis estudios de licenciatura. Por primera vez me percataba de que mi inquietud por la electrónica y la ingeniería no estaba divorciada de mis estudios de piano y que, incluso, podía ser aprovechada para generar expresiones musicales personales. ¿Si la pianola automática de Trimpin era capaz de tocar con una precisión dinámica, con velocidades y con densidades imposibles para un ser humano, no sería posible llevar estas posibilidades sonoras a la ejecución en vivo? Fue en este momento cuando surgió la idea de diseñar un equipo electrónico que posibilitara a un pianista controlar diferentes procesos musicales desde su instrumento.

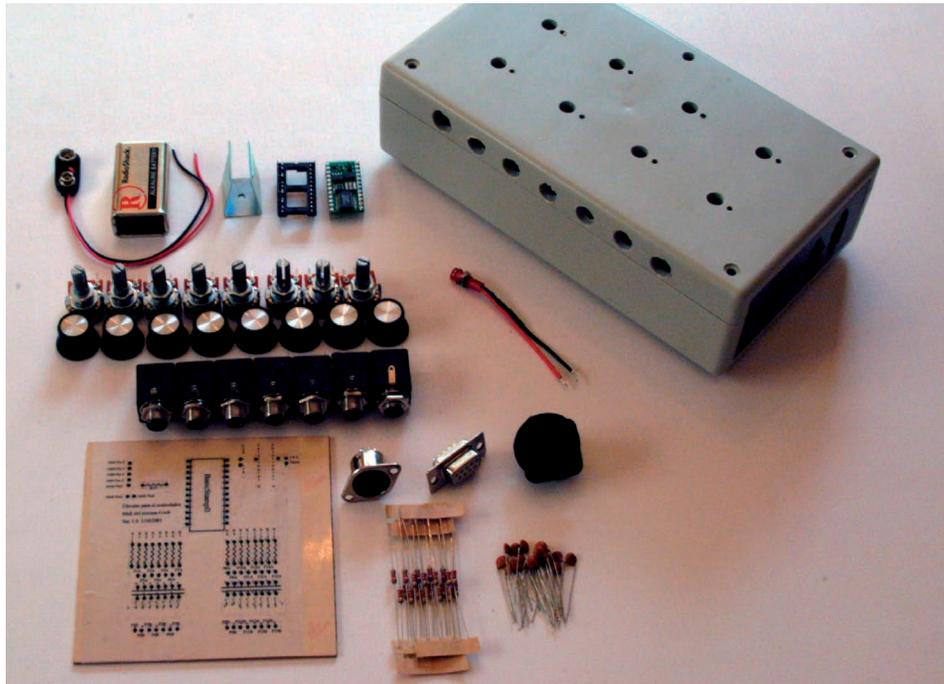


Figura 15.1. | Materiales del controlador GAB.

Mis estudios finales dentro del Laboratorio de Creación Musical, mi participación dentro del proyecto Innovación Tecnológica para la Composición Musical, una estancia en la Universidad de Nueva York en donde tuve la oportunidad de tomar cursos de programación y algunas pláticas con el matemático Miller Puket –diseñador de los programas MAX y Pure Data–, fueron las actividades que dieron forma y solidez a la idea de crear un equipo electrónico que posibilitara a un pianista controlar diferentes variables acústicas durante la ejecución de improvisaciones musicales. Así surgió el sistema GAB, un sistema diseñado con la expresa intención de permitirle a un pianista manipular en tiempo real diferentes variables musicales como son la densidad, la velocidad en la repetición de las notas, el rango dinámico, etcétera.

En esta primera versión GAB está conformado por los siguientes módulos:

- El procesador de datos es un programa de cómputo encargado de modificar, de acuerdo con los valores dados con el controlador MIDI GAB, la información registrada en el

piano para posteriormente mandar esta información procesada a un sintetizador o a un piano MIDI. El programa fue escrito en el lenguaje de programación Java aprovechando los paquetes JMSL desarrollados por Nick Didkovsky y Phil Burk, y el Midi-Share, desarrollado en el Laboratorio de Investigación de Música por Computadora del Centro Nacional de Creación Musical GRAME, en Lyon Francia.

- El Controlador MIDI GAB es un dispositivo que, conectado a una computadora, permite modificar las diferentes variables de cambio por medio de perillas y pedales. Los diferentes valores que adquieran las perillas y los pedales tendrán un resultado acústico reconocible.

La versión final del sistema GAB incluirá otros dos módulos:

- El lector MIDI para pianos acústicos permitirá leer el material ejecutado en un piano acústico y traducirlo a valores MIDI para así poder realizar los procesos de transformación en el procesador de datos.
- El reproductor automático para pianos acústicos será un equipo electrónico capaz de activar –por medio de solenoides– las teclas de un piano acústico de acuerdo con los valores MIDI que reciba en su puerto de entrada. Este equipo ofrecerá la posibilidad mecánica de alcanzar rangos dinámicos, velocidades y densidades extremas.

Es importante mencionar que el sistema GAB se encuentra en una primera fase de desarrollo. Si bien es cierto que la meta final contempla que todos los módulos y equipos que lo componen sean diseñados y construidos personalmente, dicho objetivo debe ser visto como un proyecto a largo plazo que excede en tiempo y conocimientos lo alcanzable en los estudios de licenciatura. En esta primera versión del sistema GAB el lector MIDI para pianos acústicos es sustituido por un equipo comercial y el reproductor automático para pianos acústicos es sustituido por un sintetizador.

En el presente trabajo se explica el funcionamiento del sistema GAB. Respecto al procesador de datos se explican sus posibilidades musicales y se expone y detalla su código fuente. Con el propósito de dar claridad al documento se presenta una breve explicación del

lenguaje Java y de los paquetes JMSL y MidiShare. En relación con el Controlador MIDI se explica su construcción y funcionamiento, se presenta y detalla el código fuente del microcontrolador y se da una breve explicación sobre microcontroladores, en particular sobre el BasicStamp II.

15.2 EL SISTEMA GAB

GAB es un sistema de reinterpretación musical. Su objetivo es proporcionarle al pianista una herramienta más en su trabajo creativo. Los resultados acústicos finales dependen en gran medida del material que el pianista le proporcione al sistema. GAB no pretende ser un generador de música aleatoria con procesos autónomos, por lo contrario, GAB busca ser un equipo que permita la reinterpretación de material musical. GAB puede ser visto como un ensamble de pianistas virtuales que acompañan a un pianista principal. El material que ejecutan estos “pianistas” es derivado en su totalidad del material que interpreta el pianista principal en un piano con capacidades MIDI. El nivel de similitud o diferencia entre el nuevo material respecto al material original estará determinado por el valor que tengan en ese momento las ocho variables que GAB utiliza para generar su material. Cada vez que el pianista toca una nota, GAB genera una o varias estructuras que llamaremos objeto nota. Las características de cada objeto nota estarán determinadas por los valores que en ese momento presenten las ocho variables que GAB maneja. Estas variables son:

- Número de repeticiones: permite determinar la cantidad de veces que una misma nota es repetida periódicamente.
- Tempo de repetición: permite determinar la velocidad con la que se realizan las repeticiones periódicas definidas con el número de repeticiones.
- Nivel de variación rítmica: permite determinar la exactitud en el tempo de las repeticiones de cada nota. Niveles bajos producirán pulsos regulares y niveles altos producirán pulsos con irregularidades rítmicas.
- Nivel de variación de registro: permite definir la distancia máxima posible (en número de octavas) entre la nota que toca el pianista y el objeto nota generado por GAB.

- Nivel de disonancia: permite determinar el tipo de intervalos posibles entre la nota que toca el pianista y el objeto nota generado por el sistema.
- Nivel de densidad: permite definir la cantidad de objetos nota que serán creados por cada nota que sea tocada por el pianista.
- Variación de intensidad: permite determinar la diferencia de intensidades posibles entre la nota que toca el pianista y el objeto nota generado por GAB.
- Duración de las notas: permite determinar la duración de las notas que conforman un objeto nota.

Algunas de las características de los objetos nota están previamente definidas y no son modificables; otras, como la locación de un objeto nota, son escogidas con procesos aleatorios realizados en tiempo real.

En esta primera versión el sistema GAB está conformado por dos módulos. El procesador de datos GAB es un programa de computadora en donde es posible observar el valor de las ocho variables, y es el encargado de recibir la información del pianista (vía MIDI) y los valores del controlador MIDI GAB (vía MIDI). Con dicha información el programa realiza los diferentes cálculos matemáticos necesarios para generar el nuevo material musical el cual es mandado por el puerto de salida MIDI OUT de la computadora.

El controlador MIDI GAB es un equipo electrónico encargado de convertir el valor de hasta 15 potenciómetros (para esta versión sólo se usan ocho) en información MIDI. El controlador MIDI GAB es un equipo que posibilita al pianista modificar de una forma cómoda las variables del sistema y puede ser colocado cerca del teclado del piano.

La versión actual del sistema GAB está diseñada para generar el material reinterpretado en seis canales MIDI. Para aprovechar los efectos de espacialización sonora se sugiere que estos seis pianistas virtuales estén dispuestos en una sala de conciertos de acuerdo con el esquema de la figura 15.2.

Para poder utilizar el sistema GAB se requiere:

- Una computadora personal PC o Mac con dos puertos MIDI de entrada y uno de salida con el sistema GAB correctamente instalado.

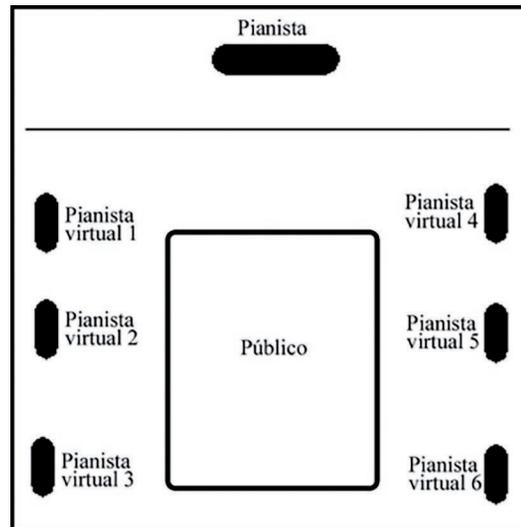


Figura 15.2. | Distribución de pianista virtuales en GAB.

- Un piano electrónico MIDI o un piano acústico con implementación MIDI.
- Seis pianos electrónicos o un sintetizador multicanal con sonido de piano y seis salidas de audio independientes.
- En caso de usar un sintetizador es necesario un sistema de amplificación de seis canales.

EL PROCESADOR DE DATOS GAB 1.0

Una vez que el procesador de datos aparece en pantalla (figura 15.3) es necesario presionar el botón Prender para activar los puertos MIDI. No es necesario apagar el sistema pues esto se realiza automáticamente al cerrar el programa. Cuando el programa es iniciado los valores de las ocho variables se establecen con su valor inicial. En la tabla 15.1 se presentan el valor inicial y los valores mínimo y máximo permitidos.



Figura 15.3. | Pantalla del procesador de datos.

Tabla 15.1

<i>Controlador</i>	<i>Valor inicial</i>	<i>Valor mínimo</i>	<i>Valor máximo</i>
Número de repeticiones	0 rep.	0 rep.	10 rep.
Tempo de repetición	.05 seg.	.05 seg.	5 seg.
Nivel de variación rítmica	0	0	10
Nivel de variación de registro	0 octavas	0 octavas	7 octavas
Nivel de disonancia	0	0	11
Nivel de densidad	1 nota	1 nota	6 notas
Variación de intensidad	0	0	127
Duración de las notas	.01 seg.	.01 seg.	5 seg.

Al mover cada una de las perillas del Controlador MIDI GAB se podrá ver el cambio en su respectivo valor numérico en pantalla y una línea de asteriscos debajo del nombre cuya dimensión irá en incremento o decremento según la posición de la perilla. Dicha línea representa gráficamente el valor actual de la perilla correspondiente.

Las variables número de repeticiones, tempo de repetición y nivel de variación rítmica

Llamaremos *objeto nota* a un conjunto de notas con alturas, intensidades, duraciones y locaciones iguales generadas a partir de una sola nota del piano principal. Por tanto, cuando hablemos de la altura, la duración, la intensidad o la locación de un *objeto nota* nos estaremos refiriendo a la altura, la duración, la intensidad o la locación de cada una de las notas que conforman dicho objeto. Cada vez que el pianista toca una nota en el piano, el procesador de datos genera uno o varios *objeto nota* (dependiendo del *nivel de densidad*). La cantidad de notas que conformen un *objeto nota* está dada por la variable *número de repeticiones*. Si esta variable presenta un valor de cero, entonces el *objeto nota* estará integrado por una sola nota que sonará en el mismo momento que la nota del piano principal (cero repeticiones).

Si la variable *número de repeticiones* presenta un valor de cero, el cambio en las variables *tempo de repetición* y *nivel de variación rítmica* no tiene efecto. Por lo contrario, si la variable *número de repeticiones* es mayor que cero, entonces la variable *tempo de repetición* indica la velocidad a la que se realizarán las repeticiones, y la variable *nivel de variación rítmica* nos permitirá definir la precisión rítmica de dichas repeticiones. Niveles bajos de *variación rítmica* generarán pulsos regulares; a medida que el nivel de *variación rítmica* se incrementa, el ritmo se vuelve cada vez más irregular.

El índice máximo de *variación rítmica* definido en el programa se fijó en cuatro unidades por lo que el nivel máximo de *variación rítmica* (127) modifica al azar el valor establecido en la variable *tempo de repetición* en un rango que puede ir de una cuarta parte de su valor, hasta cuatro veces su valor (tabla 15.2).

Tabla 15.2

<i>Índice de variación rítmica</i>	<i>Valor mínimo posible</i>	<i>Valor máximo posible</i>
0	del valor (sin modificación)	1 vez el valor (sin modificación)
64	del valor	2 veces el valor
127	del valor	4 veces el valor

Un *objeto nota* creado con los valores de la tabla 15.3 tendría las características del ejemplo mostrado en la figura 15.4.¹

Tabla 15.3

<i>Controlador</i>	<i>Valor</i>	<i>Controlador</i>	<i>Valor</i>
Número de repeticiones	0 rep.	Nivel de disonancia	0
Tempo de repetición	1.0 seg.	Nivel de densidad	1 nota
Nivel de variación rítmica	0	Variación de intensidad	0
Nivel de variación de registro	0 Octavas	Duración de las notas	1.0 seg.



Figura 15.4. | Ejemplo A de objeto generado.

Si modificamos el valor de la variable *número de repeticiones* de acuerdo con la tabla 15.4 el resultado sonoro sería el del ejemplo de la figura 15.5:

Tabla 15.4

<i>Controlador</i>	<i>Valor</i>	<i>Controlador</i>	<i>Valor</i>
Número de repeticiones	3 rep.	Nivel de disonancia	0
Tempo de repetición	1.0 seg.	Nivel de densidad	1 NOTA
Nivel de variación rítmica	0	Variación de intensidad	0
Nivel de variación de registro	0 OCTAVAS	Duración de las notas	1.0 SEG.



Figura 15.5. | Ejemplo B de objeto generado.

¹ Recordemos que la locación (canal MIDI de salida) es escogida al azar de entre los primeros seis canales MIDI. Una vez definida la locación el objeto nota se toca en dicho lugar. Por tal motivo y para fines de simplicidad sólo se utilizarán dos sistemas para exponer cada ejemplo: uno para el pianista y otro para el “pianista virtual” que podrá ser uno de entre los seis “pianistas virtuales”.

Si mantenemos el *número de repeticiones* en tres y el *nivel de variación rítmica* en cero pero modificamos el *tempo de repetición* de acuerdo con los valores de la tabla 15.5, la velocidad en las repeticiones cambiará como se puede apreciar en la figura 15.6.

Tabla 15.5

<i>Controlador</i>	<i>Valor</i>	<i>Controlador</i>	<i>Valor</i>
Número de repeticiones	3 rep.	Nivel de disonancia	0
Tempo de repetición	0.25 seg.	Nivel de densidad	1 nota
Nivel de variación rítmica	0	Variación de intensidad	0
Nivel de variación de registro	0 octavas	Duración de las notas	.25 seg.



Figura 15.6. | Ejemplo c de objeto generado.

Como ya se ha explicado, el *nivel de variación rítmica* permite modificar la precisión rítmica de las repeticiones; el ejemplo de la figura 15.7 muestra los resultados de utilizar los valores de la tabla 15.6:

Tabla 15.6

<i>Controlador</i>	<i>Valor</i>	<i>Controlador</i>	<i>Valor</i>
Número de repeticiones	10 rep.	Nivel de disonancia	0
Tempo de repetición	0.25 seg.	Nivel de densidad	1 nota
Nivel de variación rítmica	4	Variación de intensidad	0
Nivel de variación de registro	0 octavas	Duración de las notas	.01 seg.



Figura 15.7. | Ejemplo d de objeto generado.

Nivel de variación de registro

En los ejemplos anteriores hemos visto que la altura de todos los *objetos nota* ha sido la misma que la tocada en el piano principal. Esto se debe a que las variables *nivel de variación de registro* y *nivel de disonancia* se han mantenido en cero. La variable *nivel de variación de registro* permite que la altura de un objeto nota se encuentre en una octava diferente de la nota original. La variable *nivel de variación de registro* define la cantidad máxima posible de octavas entre la nota original y el objeto nota. Un proceso aleatorio define, dentro del rango establecido, la cantidad de octavas ascendentes o descendentes que habrá entre la nota original y el objeto nota. El ejemplo de la figura 15.8 muestra los resultados de utilizar los valores de la tabla 15.7:

Tabla 15.7

Controlador	Valor	Controlador	Valor
Número de repeticiones	0 rep.	Nivel de disonancia	0
Tempo de repetición	1.0 seg.	Nivel de densidad	1 nota
Nivel de variación rítmica	0	Variación de intensidad	0
Nivel de variación de registro	3 octavas	Duración de las notas	1.0 seg.



Figura 15.8. | Ejemplo E de objeto generado.

Nivel de disonancia

Así como la variable *nivel de variación de registro* se encarga de modificar la distancia en octavas, entre la nota del piano y el objeto nota, la variable *nivel de disonancia* se encarga de modificar el intervalo entre ambas notas. Nuevamente, un proceso aleatorio se encarga de definir el intervalo entre las notas. El valor del *nivel de disonancia* permite establecer el tipo de intervalos posibles. En la tabla 15.8 se observa cómo, a medida que el valor de disonancia aumenta, se van agregando posibles intervalos al proceso de selección. El intervalo entre la nota del piano y el objeto nota siempre será ascendente.

Tabla 15.8

<i>Nivel de disonancia</i>	<i>Intervalos posibles</i>
0	Unísono
1	Unísono, 3m
2	Unísono, 3m, 3m
3	Unísono, 3m, 3m, 6m
4	Unísono, 3m, 3m, 6m, 6m
5	Unísono, 3m, 3m, 6m, 6m, 5j
6	Unísono, 3m, 3m, 6m, 6m, 5j, 4j
7	Unísono, 3m, 3m, 6m, 6m, 5j, 4j, 2m
8	Unísono, 3m, 3m, 6m, 6m, 5j, 4j, 2m, 2m
9	Unísono, 3m, 3m, 6m, 6m, 5j, 4j, 2m, 2m, 7m
10	Unísono, 3m, 3m, 6m, 6m, 5j, 4j, 2m, 2m, 7m, 7m
11	Unísono, 3m, 3m, 6m, 6m, 5j, 4j, 2m, 2m, 7m, 7m, 4a

Veamos cómo los valores de la tabla ;15.9. producen intervalos de unísono, tercera menor, tercera mayor, sexta menor, sexta mayor y quinta justa (figura 15.9):

Tabla 15.9

<i>Controlador</i>	<i>Valor</i>	<i>Controlador</i>	<i>Valor</i>
Número de repeticiones	0 rep.	Nivel de disonancia	5
Tempo de repetición	1.0 seg.	Nivel de densidad	1 nota
Nivel de variación rítmica	0	Variación de intensidad	0
Nivel de variación de registro	0	Duración de las notas	1.0 seg.

octavas

The figure shows a musical score for Piano and Guitarrón. The Piano part is in the upper staff and the Guitarrón part is in the lower staff. Both parts play a sequence of notes. Below the notes, there are interval labels: 3M, 0, 6m, 3m, 3m, 6M, 5J, 0. These labels correspond to the intervals between consecutive notes in the sequence.

Figura 15.9. | Ejemplo F de objeto generado.

Nivel de densidad

Hasta este momento, todos los ejemplos que se han visto han contemplado un *nivel de densidad* de uno. Con este valor únicamente se crea un objeto nota por cada nota del piano principal. Sin embargo, es posible generar más de un objeto nota por cada nota del piano principal. La variable *nivel de densidad* establece la cantidad de objetos nota que deberán crearse por cada nota. Recordemos que la locación de cada objeto nota se establece de forma aleatoria. Veamos la figura 15.10, resultado de utilizar los valores de la tabla 15.10.

Tabla 15.10

Controlador	Valor	Controlador	Valor
Número de repeticiones	3 rep.	Nivel de disonancia	0
Tempo de repetición	1.0 seg.	Nivel de densidad	3 notas
Nivel de variación rítmica	0	Variación de intensidad	0
Nivel de variación de registro	0 octavas	Duración de las notas	1.0 seg.

The figure shows a musical score with seven staves. The top staff is labeled 'Piano P' and contains three measures, each with a single quarter note. Below it are six staves labeled 'PV 1' through 'PV 6'. Each of these six staves contains three measures, each with three quarter notes. This illustrates a density of 3 notes per piano note.

Figura 15.10. | Ejemplo G de objeto generado.

Variación de intensidad

El procesador de datos permite modificar la intensidad de los objetos nota. Si la variable presenta un valor de cero, los objetos nota se crean con la misma intensidad que la nota del

piano principal. A medida que el valor de la variable *variación de intensidad* aumenta, la diferencia de intensidad entre la nota original y el objeto nota es mayor. Un proceso aleatorio aumenta o disminuye la intensidad del objeto nota respecto a la nota original según el valor de esta variable. En la figura 15.11 se puede ver cómo la intensidad de las notas generadas por el sistema es medianamente diferente de las notas del piano principal. Los valores de las variables para este ejemplo se muestran en la tabla 15.11.

Tabla 15.11

<i>Controlador</i>	<i>Valor</i>	<i>Controlador</i>	<i>Valor</i>
Número de repeticiones	1 rep.	Nivel de disonancia	0
Tempo de repetición	1.0 seg.	Nivel de densidad	1 nota
Nivel de variación rítmica	0	Variación de intensidad	64
Nivel de variación de registro	0 octavas	Duración de las notas	1.0 seg.



Figura 15.11. | Ejemplo H de objeto generado.

Duración de las notas

Hasta este momento y por fines de notación en los ejemplos, la duración de las notas (tanto en el piano principal, como en los *objetos nota*) se ha mantenido de forma esquemática. El procesador de datos GAB no reconoce la duración de una nota tocada por el pianista, únicamente recibe la información del momento en que es presionada. Por tal motivo, la duración de las notas de un objeto nota es siempre fija y está definida por la variable *duración de las notas*. El rango para esta variable va de 0.01 a 5.0 segundos. En la figura 15.12 se puede ver que independientemente de la duración de la nota principal las notas generadas por el sistema mantienen una duración constante. La tabla 15.12 muestra los valores utilizados para este ejemplo.

Tabla 15.12

<i>Controlador</i>	<i>Valor</i>	<i>Controlador</i>	<i>Valor</i>
Número de repeticiones	2 rep.	Nivel de disonancia	0
Tempo de repetición	1.0 seg.	Nivel de densidad	1 nota
Nivel de variación rítmica	0	Variación de intensidad	0
Nivel de variación de registro	0 octavas	Duración de las notas	.5 seg.

**Figura 15.12.** | Ejemplo 1 de objeto generado.

JMSL Y MIDISHARE

JMSL (Java Music Specification Language) es una herramienta diseñada en Java enfocada al diseño de composiciones con algoritmos e instrumentos inteligentes con aplicaciones en la ejecución musical en vivo. JMSL es el sucesor del HMSL (Hierarchical Music Specification Language), lenguaje, este último, basado en FORTH y diseñado en el Mills College Center for Contemporary Music por Phil Burk, Larry Polansky y David Rosenboom. En 1997, al percatarse de que la computadora Amiga quedaría fuera del mercado, y con el surgimiento de un lenguaje que prometía funcionar independientemente de la plataforma, Nick Didkovsky inició la exportación de HMSL a Java. Finalmente, al ver la gran ventaja que ofrecía JMSL, el mismo Didkovsky y el coautor de HMSL, Phil Burk, no sólo exportaron HMSL a Java sino que aprovecharon las posibilidades de este último para mejorar y simplificar HMSL. JMSL amplía a Java mediante una gran cantidad de clases con aplicaciones musicales: temporalización jerárquica de objetos musicales, generación de secuencias, funciones de distribución, etc. Estas clases ofrecen un territorio muy amplio para la exploración musical.

La solidez de JMSL radica en su derivación de Java lo que lo hace un sistema abierto al propio desarrollo de Java incluyendo su conectividad a bases de datos, a las herramientas propias de las redes, a los paquetes gráficos y a numerosos paquetes en constante implementación. Actualmente JMSL soporta los paquetes MidiPort de Robert Marsanyi, MidiShare de GRAME y JSyn de Phil Burk.

La noción de jerarquía es el punto central de JMSL. Una jerarquía es una simple colección de relaciones padre-hijo. Un padre podría ser, por ejemplo, una sinfonía. El primero, el segundo y el tercer movimientos serían los hijos. Como el segundo movimiento no inicia sino hasta terminar el primer movimiento, y el tercer movimiento no inicia sino hasta terminar el segundo, este es un caso de una colección secuencial de eventos. Por otro lado, un ejemplo de colección paralela sería el objeto cuarteto de cuerdas, en este caso los hijos: violín primero, violín segundo, viola y chelo deben iniciar al mismo tiempo y mantenerse en constante sincronía.

JMSL permite agendar eventos arbitrarios en el tiempo (musicales y no musicales), todos estos “trabajos musicales” se pueden agrupar en colecciones y, a su vez, las colecciones pueden ser agrupadas en supercolecciones. Finalmente las supercolecciones, las colecciones y cualquier “trabajo musical” puede ser agendado en el tiempo, ya sea paralela o secuencialmente, respecto a otras colecciones.

JMSL ofrece también clases para generar, modificar y acceder a listados musicales (MusicShapes) que son arreglos numéricos de dos dimensiones equivalentes a tablas con un número cualquiera de renglones y columnas. Dichos arreglos pueden ser utilizados e interpretados por un trabajo musical de muchas formas diversas, por ejemplo, para definir la duración de un evento, intensidad, número de repeticiones, índices de variación tímbrica, procesos estocásticos, etcétera.

Como se puede apreciar, JMSL es una herramienta muy útil en los campos de la composición y la ejecución musical. Permite diseñar y solucionar ideas musicales complejas y, al ser un sistema abierto, permite relacionar eventos musicales con otro tipo de estructuras.

Por su parte, MidiShare 1.86 fue desarrollado en el Laboratorio de Investigación de Música por Computadora del Centro Nacional de Creación Musical GRAME en Lyon, Francia. Al igual que JMSL, MidiShare amplía a Java mediante un grupo de clases con aplicaciones en el uso del protocolo MIDI. Con las clases de MidiShare es posible acceder a los puertos MIDI de una computadora y realizar una gran cantidad de operaciones con la información obtenida.

EXPLICACIÓN Y TRANSCRIPCIÓN DEL CÓDIGO FUENTE

El programa GAB está compuesto de tres clases: i) la clase *Gab*, que es la clase principal, extendida de *Applet* y que implementa dos interfases: *MidiListener*, para poder recibir MIDI; y *ActionListener*, para poder responder al uso del botón *Prender*-; ii) la clase *NotaCompuesta* extendida de *MusicJob* encargada de generar la estructura de los *objetos nota*; y iii) la clase *NotaBase* encargada de generar las notas que conforman los *objetos nota*. Esta última implementa la interfase *Playable* con el fin de poder ser utilizada en los métodos de agenda de la clase *NotaCompuesta*.

Al principio de la clase *Gab* se definen e inicializan todas las variables necesarias para el funcionamiento del programa: variables numéricas, arreglos, etiquetas, botones y tipos de letra. En el método *Main* se definen los valores necesarios para poder ejecutar el programa como una aplicación fuera de las páginas web. En el método *start* se crean y agregan el título, el botón y todas las etiquetas a la interfase para el usuario del programa. El método *inicio* es invocado cada vez que el botón *Prender* es presionado y se encarga de prender los puertos MIDI. En el método *cambiaGráficos* se encuentran todas las rutinas necesarias para poder actualizar los valores y líneas de asteriscos en la interfase al usuario cada vez que es recibido el mensaje MIDI correspondiente. El método *actionPerformed* se encarga de definirle al botón *Prender* su capacidad de prender las rutinas MIDI.

Los métodos *handleNoteOn*, *handleNoteOff*, *handlePolyphonicAftertouch*, *handleControlChange*, *handleProgramChange*, *handleChannelAftertouch* y *handlePitchBend* son los métodos encargados de recibir la información MIDI proveniente del Puerto MIDI IN de la computadora. Todos estos métodos, a excepción de *handleControlChange* y *handleNoteOn*, se encargan de mandar sin modificación la información que reciben en su entrada al puerto de salida MIDI OUT de la computadora.

Los métodos *handleNoteOn* y *handleControlChange* son los métodos MIDI (implementados por *MidiListener*) importantes en el programa. El primero se encarga de invocar al método *alTocarNota* que se describirá más adelante. El segundo se encarga de actualizar las variables que generan los valores de los ocho parámetros que GAB puede modificar. Los números de

ControlChange que el programa reconoce para realizar los procesos se presentan en la tabla 15.13.²

Tabla 15.13

<i>Parámetros en GAB</i>	<i>Número de Control Change</i>
Número de repeticiones	16
Tempo de repetición	17
Nivel de variación rítmica	18
Nivel de variación de registro	19
Nivel de disonancia	20
Nivel de densidad	21
Variación de intensidad	22
Duración de las notas	23

Cada vez que una nota es recibida desde el piano principal se crea y activa un objeto *NotaCompuesta* utilizando el valor de altura e intensidad de dicha nota. El método *alTocarNota* es el encargado directo de generar todos los *objetos nota*. La clase *NotaCompuesta* es el molde o marco de las estructuras *objeto nota*. Esta clase utiliza la *notaBase* y los valores de *número de repeticiones* y *tempo de repeticiones* para generar las características particulares de cada *objeto nota*. Además, esta clase utiliza el valor de *nivel de densidad* para decidir la cantidad de *objetos nota* que deberán ser creados. La clase *NotaBase* es el molde de las notas que conforman un *objeto nota*, en dicha clase están la mayoría de los métodos utilizados para generar la altura, la intensidad, la duración y la locación de los *objetos nota* según el valor que en ese momento presenten los parámetros de *nivel de variación de registro*, *nivel de disonancia*, *variación de intensidad* y *duración de las notas*.

A continuación se transcribe íntegro y sin modificaciones el código fuente del programa:

1. */* Procesador de Datos GAB*
2. *versión 1.0 autor Hugo Solís Fecha 30 de octubre*
3. *del 2001 Al recibir Notas MIDI en el puerto de*
4. *entrada 0, el programa genera notas MIDI en el*
5. *puerto de salida 0 con diferentes características*
6. *de acuerdo a los valores de los mensajes Control*

² Los valores 16 a 23 fueron escogidos debido a que en las especificaciones MIDI son valores para propósitos generales y no interfieren con las funciones de los equipos comerciales.

```

7. Change 16 a 23 que reciba. El programa genera un
8. applet con un botón para activar MIDI y permite
9. visualizar los valores de las ocho variables que
10. utiliza para generar las Notas. Las variables
11. son: Número de repeticiones, Tempo de repetición,
12. Nivel de variación rítmica, Nivel de variación de
13. registro, Nivel de disonancia, Nivel de densidad
14. Variación de intensidad y Duración de las notas.
15. Este procesador está diseñado para ser usado con
16. el Controlador MIDI GAB Para detalles de
17. funcionamiento ver la tesis «Gab 1.0, sistema de
18. reinterpretación electrónica para pianos
19. acústicos» de Hugo Solís */
20.
21. package gabver1;
22.
23. import java.awt.*;
24. import java.awt.event.*;
25. import com.softsynth.jmsl.*;
26. import com.softsynth.jmsl.midi.*;
27.
28. public class Gab extends java.applet.Applet
29. implements MidiListener, ActionListener
30. {
31. //variables de inicialización
32. int numDeCanales = 6;
33. double durMaxDeNota = 5;
34. double durMinDeNota = .01;
35. int maxDeDensidad = 6;
36. int numMaxDeRepeticiones = 10;
37. double tiempoMinimoDeRepeticion = .05;
38. double tiempoMaximoDeRepeticion = 5.0;
39. double unidadDeVariacionRitmica = 4.0;
40.
41. //variables generales
42. int varRep = 0;
43. int varTem = 0;
44. int varRit = 0;
45. int varReg = 0;
46. int varDis = 0;
47. int varDen = 0;
48. int varInt = 0;
49. int varDur = 0;
50. int dis[] = {0,4,3,9,8,7,5,2,1,10,11,6};
51. int altura;
52. int intensidad;
53. MidiParser midiFrases;
54.
55. //Etiquetas, botones y tiposdeletra
56. Label etiTitulo;
57. Label etiLinea;

```

```
58. Label etiNumDeRep;
59. Label etiTmpDeRep;
60. Label etiVarDeRit;
61. Label etiVarDeReg;
62. Label etiNivDeDis;
63. Label etiNivDeDen;
64. Label etiVarDeInt;
65. Label etiNivDeDur;
66. Label etiNumDeRepB;
67. Label etiTmpDeRepB;
68. Label etiVarDeRitB;
69. Label etiVarDeRegB;
70. Label etiNivDeDisB;
71. Label etiNivDeDenB;
72. Label etiVarDeIntB;
73. Label etiNivDeDurB;
74. Label etiNumDeRepC;
75. Label etiTmpDeRepC;
76. Label etiVarDeRitC;
77. Label etiVarDeRegC;
78. Label etiNivDeDisC;
79. Label etiNivDeDenC;
80. Label etiVarDeIntC;
81. Label etiNivDeDurC;
82. Button botonPaPrender;
83. Font titFont;
84. Font etiFont;
85.
86. // Definición de Main
87. public static void main(String args[])
88. {
89.     Gab gaby = new Gab();
90.     gaby.start();
91.     gaby.setSize(1100,1100);
92.     gaby.setVisible(true);
93.     Frame f= new
94.     Frame("Interfase del Sistema GAB 1.0");
95.     f.addWindowListener(
96.     new java.awt.event.WindowAdapter()
97.     {
98.     public void windowClosing(
99.     java.awt.event.WindowEvent e)
100.    {
101.        JMSL.midi.closeDevices();
102.        System.exit(0);
103.    }
104.    });
105.     f.add(gaby, BorderLayout.CENTER);
106.     f.setSize(1100,1025);
107.     f.setVisible(true);
108. }
```

```
109.
110. //Definición del método principal Gab
111. public void start()
112. {
113.     setLayout (new BorderLayout());
114.     //Definición de tipos de letra,
115.     //etiquetas y paneles
116.     Font titFont = new Font("TimesRoman",
117.     Font.BOLD, 48);
118.     Font etiFont = new Font("TimesRoman",
119.     Font.BOLD, 30);
120.
121.     etiTitulo = new Label("Procesador de
122.     Datos GAB 1.0", Label.CENTER);
123.     etiTitulo.setFont(titFont);
124.     botonPaPrender = new Button("Prender");
125.     etiLinea = new Label("-----", Label.CENTER);
126.     etiLinea.setFont(etiFont);
127.     etiNumDeRep = new Label(
128.     "Número de repeticiones", Label.LEFT);
129.     etiNumDeRep.setFont(etiFont);
130.     etiTmpDeRep = new Label(
131.     "Tempo de repetición", Label.LEFT);
132.     etiTmpDeRep.setFont(etiFont);
133.     etiVarDeRit = new Label(
134.     "Nivel de variación rítmica", Label.LEFT);
135.     etiVarDeRit.setFont(etiFont);
136.     etiVarDeReg = new Label(
137.     "Nivel de variación de registro", Label.LEFT);
138.     etiVarDeReg.setFont(etiFont);
139.     etiNivDeDis = new Label(
140.     "Nivel de disonancia", Label.LEFT);
141.     etiNivDeDis.setFont(etiFont);
142.     etiNivDeDen = new Label(
143.     "Nivel de densidad", Label.LEFT);
144.     etiNivDeDen.setFont(etiFont);
145.     etiVarDeInt = new Label(
146.     "Variación de intensidad", Label.LEFT);
147.     etiVarDeInt.setFont(etiFont);
148.     etiNivDeDur = new Label(
149.     "Duración de las notas", Label.LEFT);
150.     etiNivDeDur.setFont(etiFont);
151.     etiNumDeRepB = new Label(
152.     "0", Label.LEFT);
153.     etiNumDeRepB.setFont(etiFont);
154.     etiTmpDeRepB = new Label(
155.     Double.toString(tiempoMinimoDeRepeticion),
156.     Label.LEFT);
157.     etiTmpDeRepB.setFont(etiFont);
158.     etiVarDeRitB = new Label(
159.     "0", Label.LEFT);
```

```
160. etiVarDeRitB.setFont(etiFont);
161. etiVarDeRegB = new Label(
162. "0", Label.LEFT);
163. etiVarDeRegB.setFont(etiFont);
164. etiNivDeDisB = new Label(
165. "0", Label.LEFT);
166. etiNivDeDisB.setFont(etiFont);
167. etiNivDeDenB = new Label(
168. "1", Label.LEFT);
169. etiNivDeDenB.setFont(etiFont);
170. etiVarDeIntB = new Label(
171. "0", Label.LEFT);
172. etiVarDeIntB.setFont(etiFont);
173. etiNivDeDurB = new Label(
174. Double.toString(durMinDeNota), Label.LEFT);
175. etiNivDeDurB.setFont(etiFont);
176. etiNumDeRepC = new Label("**", Label.LEFT);
177. etiNumDeRepC.setFont(etiFont);
178. etiTmpDeRepC = new Label("**", Label.LEFT);
179. etiTmpDeRepC.setFont(etiFont);
180. etiVarDeRitC = new Label("**", Label.LEFT);
181. etiVarDeRitC.setFont(etiFont);
182. etiVarDeRegC = new Label("**", Label.LEFT);
183. etiVarDeRegC.setFont(etiFont);
184. etiNivDeDisC = new Label("**", Label.LEFT);
185. etiNivDeDisC.setFont(etiFont);
186. etiNivDeDenC = new Label("**", Label.LEFT);
187. etiNivDeDenC.setFont(etiFont);
188. etiVarDeIntC = new Label("**", Label.LEFT);
189. etiVarDeIntC.setFont(etiFont);
190. etiNivDeDurC = new Label("**", Label.LEFT);
191. etiNivDeDurC.setFont(etiFont);
192.
193. Panel n = new Panel();
194. n.setLayout(new BorderLayout());
195. n.add("Center", etiTitulo);
196. n.add("East", botonPaPrender);
197. n.add("South", etiLinea);
198.
199. Panel c = new Panel();
200. c.setLayout(new GridLayout(8,1,0,25));
201.
202. Panel c1 = new Panel();
203. c1.setLayout(new BorderLayout());
204. c1.add("Center", etiNumDeRep);
205. c1.add("East", etiNumDeRepB);
206. c1.add("South", etiNumDeRepC);
207. Panel c2 = new Panel();
208. c2.setLayout(new BorderLayout());
209. c2.add("Center", etiTmpDeRep);
210. c2.add("East", etiTmpDeRepB);
```

```
211. c2.add("South", etiTmpDeRepC);
212. Panel c3 = new Panel();
213. c3.setLayout(new BorderLayout());
214. c3.add("Center", etiVarDeRit);
215. c3.add("East", etiVarDeRitB);
216. c3.add("South", etiVarDeRitC);
217. Panel c4 = new Panel();
218. c4.setLayout(new BorderLayout());
219. c4.add("Center", etiVarDeReg);
220. c4.add("East", etiVarDeRegB);
221. c4.add("South", etiVarDeRegC);
222. Panel c5 = new Panel();
223. c5.setLayout(new BorderLayout());
224. c5.add("Center", etiNivDeDis);
225. c5.add("East", etiNivDeDisB);
226. c5.add("South", etiNivDeDisC);
227. Panel c6 = new Panel();
228. c6.setLayout(new BorderLayout());
229. c6.add("Center", etiNivDeDen);
230. c6.add("East", etiNivDeDenB);
231. c6.add("South", etiNivDeDenC);
232. Panel c7 = new Panel();
233. c7.setLayout(new BorderLayout());
234. c7.add("Center", etiVarDeInt);
235. c7.add("East", etiVarDeIntB);
236. c7.add("South", etiVarDeIntC);
237. Panel c8 = new Panel();
238. c8.setLayout(new BorderLayout());
239. c8.add("Center", etiNivDeDur);
240. c8.add("East", etiNivDeDurB);
241. c8.add("South", etiNivDeDurC);
242.
243. c.add(c1);
244. c.add(c2);
245. c.add(c3);
246. c.add(c4);
247. c.add(c5);
248. c.add(c6);
249. c.add(c7);
250. c.add(c8);
251.
252. add("North", n);
253. add("Center", c);
254.
255. // Definición de receptores de acción
256. botonPaPrender.addActionListener(this);
257.
258. WindowAdapter myWindowAdapter =
259. new WindowAdapter()
260. {
261. public void windowClosing(WindowEvent e)
```

```
262. {
263.   JMSL.midi.closeDevices();
264.   System.exit(0);
265. }
266. };
267.
268. //Generador de valores aleatorios
269. JMSLRandom.randomize();
270. }
271.
272. // Definición del método inicio,
273. //se ejecuta el precionar el boton «prender»
274. public void inicio()
275. {
276.   JMSL.midi.openDevices(0,0);
277.   MidiParser midiFrases = new MidiParser();
278.   JMSL.midi.addMidiParser( midiFrases);
279.   midiFrases.addMidiListener(this);
280.   System.out.println(
281.     "Se ha iniciado el programa");
282. }
283.
284. //Definición del método cambiaGráficos,
285. //encargado de actualizar
286. //los valores en pantalla, es invocado
287. //cada que se recibe un ControlChange 16-23
288. public void cambiaGraficos(int id, int valor)
289. {
290.   StringBuffer cadena = new StringBuffer("");
291.   for(int i = 0; i < valor/2; i++)
292.   {
293.     cadena.append("*");
294.   }
295.
296.   switch(id)
297.   {
298.     case 16:
299.       etiNumDeRepB.setText(Integer.toString(
300.         (varRep * numMaxDeRepeticiones / 127)));
301.       etiNumDeRepC.setText(cadena.toString());
302.       break;
303.     case 17:
304.       etiTmpDeRepB.setText(Double.toString(
305.         new Double(new Double(100*(Math.min(
306.           (varTem * tiempoMaximoDeRepeticion / 127) +
307.           tiempoMinimoDeRepeticion,
308.           tiempoMaximoDeRepeticion)
309.         ).intValue()).doubleValue() / 100));
310.       etiTmpDeRepC.setText(cadena.toString());
311.       break;
312.     case 18:
```

```

313. etiVarDeRitB.setText(Integer.toString(
314. (varRit * 10 / 127)));
315. etiVarDeRitC.setText(cadena.toString());
316. break;
317. case 19:
318. etiVarDeRegB.setText(Integer.toString(
319. valor * 8 / 127));
320. etiVarDeRegC.setText(cadena.toString());
321. break;
322. case 20:
323. etiNivDeDisB.setText(Integer.toString(
324. valor * 11 / 127));
325. etiNivDeDisC.setText(cadena.toString());
326. break;
327. case 21:
328. etiNivDeDenB.setText(Integer.toString(
329. (valor * maxDeDensidad / 127) + 1));
330. etiNivDeDenC.setText(cadena.toString());
331. break;
332. case 22:
333. etiVarDeIntB.setText(Integer.toString(valor));
334. etiVarDeIntC.setText(cadena.toString());
335. break;
336. case 23:
337. etiNivDeDurB.setText(Double.toString(
338. new Double(new Double(100*(Math.min(
339. (varDur * durMaxDeNota / 127) +
340. durMinDeNota, durMaxDeNota))
341. ).intValue()).doubleValue() / 100));
342. etiNivDeDurC.setText(cadena.toString());
343. break;
344. }
345. }
346.
347. //Métodos de ActionListener, encargado
348. //de definir que el botón
349. //”prende” realice el método inicio al
350. //ser presionado
351. public void actionPerformed(ActionEvent e)
352. {
353. Object source = e.getSource();
354. if (source == botonPaPrender) inicio();
355.
356. }
357.
358. // Metodos de recepción MIDI, necesario
359. //definir todos al implementar MidiListener
360. public void handleNoteOn(double timeStamp,
361. int channel, int pitch, int velocity)
362. {
363. altura = pitch;

```

```
364. intensidad = velocity;
365.
366. alTocarNota();
367. }
368.
369. public void handleNoteOff(double timeStamp,
370. int channel, int pitch, int velocity)
371. {
372. }
373.
374. public void handlePolyphonicAftertouch(
375. double timeStamp,
376. int channel, int pitch, int pressure)
377. {
378. JMSL.midi.sendMessage(160 + (channel - 1),
379. pitch, pressure);
380. }
381.
382. public void handleControlChange(
383. double timeStamp, int channel,
384. int id, int valor)
385. {
386. switch(id)
387. {
388. case 16:
389. varRep = valor;
390. cambiaGraficos(id, valor);
391. break;
392. case 17:
393. varTem = valor;
394. cambiaGraficos(id, valor);
395. break;
396. case 18:
397. varRit = valor;
398. cambiaGraficos(id, valor);
399. break;
400. case 19:
401. varReg = valor;
402. cambiaGraficos(id, valor);
403. break;
404. case 20:
405. varDis = valor;
406. cambiaGraficos(id, valor);
407. break;
408. case 21:
409. varDen = valor;
410. cambiaGraficos(id, valor);
411. break;
412. case 22:
413. varInt = valor;
414. cambiaGraficos(id, valor);
```

```
415. break;
416. case 23:
417.   varDur = valor;
418.   cambiaGraficos(id, valor);
419.   break;
420. default:
421.   JMSL.midi.control(channel, id, valor);
422. }
423. }
424.
425. public void handleProgramChange(
426.   double timeStamp, int channel, int program)
427. {
428.   JMSL.midi.programChange(channel, program);
429. }
430.
431. public void handleChannelAftertouch(
432.   double timeStamp, int channel, int pressure)
433. {
434.   JMSL.midi.sendMessage(208 +
435.     (channel - 1), pressure, 0);
436. }
437.
438. public void handlePitchBend(
439.   double timeStamp, int channel, int lsb, int msb)
440. {
441.   JMSL.midi.sendMessage(
442.     224 + (channel - 1), lsb, msb);
443. }
444.
445. // El método al TocarNota se invoca
446. //cada que es recibido un
447. //mensaje de NoteOn. Este método
448. //genera una o varias NotasCompuestas
449. //según el valor de densidad.
450. public void alTocarNota()
451. {
452.   NotaCompuesta miMJ = new NotaCompuesta();
453.
454.   int valorDeDensidad = (varDen *
455.     maxDeDensidad / 127) + 1;
456.   for (int i = 0; i < valorDeDensidad; i++)
457.   {
458.     NotaBase miNota = new NotaBase();
459.     miMJ.addRepeatPlayable(miNota);
460.   }
461.   miMJ.launch(JMSL.now());
462. }
463.
464. // La clase NotaCompuesta genera
465. //según los valores de NumeroDeRepetición,
```

```
466. //Tempo de Repetición y Variación
467. //rítmica una estructura musical que será
468. // utilizada por alTocarNota
469. public class NotaCompuesta extends MusicJob
470. {
471.     double tmpDeRep;
472.     double nivVarRi;
473.
474.     NotaCompuesta()
475.     {
476.         tmpDeRep = Math.min((varTem *
477.         tiempoMaximoDeRepeticion / 127) +
478.         tiempoMinimoDeRepeticion,
479.         tiempoMaximoDeRepeticion);
480.         nivVarRi = ((varRit * (
481.         unidadDeVariacionRitmica - 1) / 127) + 1);
482.         setRepeatPause(tmpDeRep);
483.         setRepeats(numeroDeRepeticiones(0));
484.     }
485.
486.     public double repeat(double playTime)
487.     throws InterruptedException
488.     {
489.         setRepeatPause(variacionRitmica(0));
490.         return (playTime);
491.     }
492.
493.     public int numeroDeRepeticiones(int nr)
494.     {
495.         nr = (varRep *
496.         numMaxDeRepeticiones / 127) + 1;
497.         return (nr);
498.     }
499.
500.     public double variacionRitmica(double barRit)
501.     {
502.         int volado;
503.         volado = JMSLRandom.choose(1);
504.         if (volado == 0) barRit = tmpDeRep *
505.         JMSLRandom.choose(1.0, nivVarRi);
506.         if (volado == 1) barRit = tmpDeRep /
507.         JMSLRandom.choose(1.0, nivVarRi);
508.         return (barRit);
509.     }
510. }
511.
512. //La clase NotaBase se encarga de
513. //generar una nota con canal, altura,
514. //intensidad y duración específica
515. // según los valores que en ese momento tengan
516. //los controladores.
```

```
517. class NotaBase implements Playable
518. {
519.   int miAltura;
520.   int miIntensidad;
521.   int miCanal;
522.   double tiempoApaga;
523.
524.   NotaBase()
525.   {
526.     miAltura = creadorDeAltura(altura);
527.     miIntensidad = creadorDeIntensidad(intensidad);
528.     miCanal = creadorDeCanal(0);
529.     tiempoApaga = creadorDeTiempoApaga(0);
530.   }
531.
532.   public double play(double playTime,
533.     Composable thing)
534.   {
535.     JMSL.midi.noteOn(miCanal, miAltura,
536.       miIntensidad);
537.     JMSL.midi.noteOff(JMSL.now() +
538.       tiempoApaga, miCanal, miAltura);
539.     return(playTime);
540.   }
541.
542.   int nivelDeDisonancia(int disonancia)
543.   {
544.     int valorDeDisonancia = varDis * 11 / 127;
545.     disonancia = dis[JMSLRandom.choose(
546.       valorDeDisonancia + 1)];
547.     return(disonancia);
548.   }
549.
550.   int nivelDeRegistro(int registro)
551.   {
552.     registro = registro +
553.       (( JMSLRandom.choosePlusMinus(varReg / 21)*12));
554.     while (registro < 0 || registro > 127)
555.       registro = registro +
556.         (( JMSLRandom.choosePlusMinus(
557.           varReg / 21)*12));
558.     return(registro);
559.   }
560.
561.   int creadorDeAltura(int altura)
562.   {
563.     int miAltura = nivelDeDisonancia(0) +
564.       nivelDeRegistro(altura);
565.     while (miAltura < 0 || miAltura > 127)
566.       miAltura =
567.         nivelDeDisonancia(0) +
```

```
568. nivelDeRegistro(altura);
569. altura = miAltura;
570. return(altura);
571. }
572.
573. int creadorDeIntensidad(int intensidad)
574. {
575. intensidad = intensidad +
576. JMSLRandom.choosePlusMinus(varInt);
577. while (intensidad < 1 || intensidad > 127)
578. intensidad =
579. intensidad +
580. JMSLRandom.choosePlusMinus(varInt);
581. return(intensidad);
582. }
583.
584. int creadorDeCanal(int canal)
585. {
586. canal = JMSLRandom.choose(1,
587. numDeCanales + 1);
588. return(canal);
589. }
590.
591. double creadorDeTiempoApaga(double tiempoApaga)
592. {
593. tiempoApaga = Math.min(
594. (varDur * durMaxDeNota / 127) +
595. durMinDeNota, durMaxDeNota);
596. return(tiempoApaga);
597. }
598. }
599.
```

EL CONTROLADOR MIDI GAB

El controlador MIDI GAB (figura 15.13) es un dispositivo electrónico que permite modificar los parámetros del procesador de datos de una manera sencilla mientras se está tocando el piano. Este controlador consta de ocho perillas y siete conectores para pedales MIDI comerciales.³ La información MIDI que el controlador manda en su puerto de salida MIDI OUT se muestra en la tabla 15.14.

³ La versión actual del controlador tiene deshabilitados los conectores de pedal pues no son utilizados por el procesador.



Figura 15.13. | Controlador GAB.

Tabla 15.14

<i>Control</i>	<i>Número de MIDI Control Change</i>	<i>Control</i>	<i>Número de MIDI Control Change</i>
Perilla 1	16 (0-127)	Pedal 1 (9)	24 (0-127)
Perilla 2	17 (0-127)	Pedal 2 (10)	25 (0-127)
Perilla 3	18 (0-127)	Pedal 3 (11)	26 (0-127)
Perilla 4	19 (0-127)	Pedal 4 (12)	27 (0-127)
Perilla 5	20 (0-127)	Pedal 5 (13)	28 (0-127)
Perilla 6	21 (0-127)	Pedal 6 (14)	29 (0-127)
Perilla 7	22 (0-127)	Pedal 7 (15)	30 (0-127)
Perilla 8	23 (0-127)	-----	-----

El empleo de este equipo es totalmente intuitivo y no requiere ninguna explicación. Este equipo es alimentado por una pila de 9 v, tiene un interruptor para prenderlo y apagarlo, un led para ver su estado de actividad y un puerto DB9 para conectarlo a una computadora personal en caso de que se quiera modificar el programa residente en el microcontrolador. La parte más importante para el funcionamiento de este equipo es el microcontrolador BasicStamp. Este circuito integrado es el encargado de convertir los valores analógicos de los potenciómetros en información digital codificada de acuerdo con los lineamientos del protocolo MIDI.

CONSTRUCCIÓN DEL CONTROLADOR MIDI

No se requiere un gran conocimiento en electrónica ni un equipo muy sofisticado para poder hacer un controlador MIDI GAB. Las ventajas de diseñar uno mismo este tipo de aparatos radica principalmente en lo económico que pueden resultar. Los materiales requeridos para la construcción de un controlador MIDI igual al controlador MIDI GAB se listan en la tabla 15.15. La figura 15.14 presenta el diagrama del circuito.

Tabla 15.15

Lista de materiales	
1	Gabinete de tamaño adecuado
1	Placa de cobre lisa o previamente perforada de 10 x 10 cm
1	Microprocesador BasicStampII de Páralax
1	Base para circuito integrado de 24 pines
1	Pila de 9 V.
1	Soporte para pila de 9 v
1	Conector para pila de 9 v
1	Led con resistencia integrada
1	Interruptor de paso
1	Conector DB9 hembra para chasis
1	Conector DIN hembra de 5 pines para chasis
8	Potenciómetros lineales de 50 K Ω
8	Perillas para potenciómetro
7	Conectores para plug estéreo
17	Resistencias de 220 Ω
15	Capacitores de .01 μ F
***	Tornillos y tuercas de diferentes tamaños

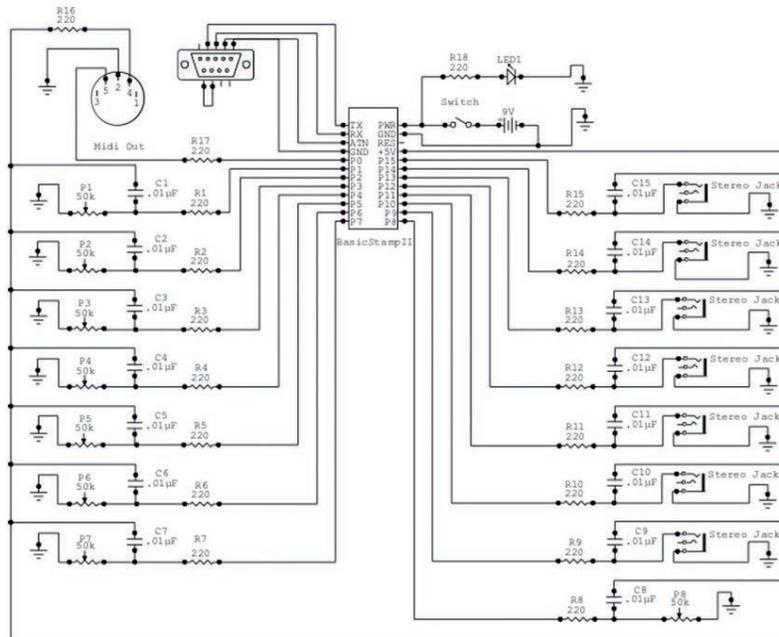


Diagrama del Controlador MIDI GAB

Versión: 1.0

Autor: Hugo Solís García

Fecha: 25 de octubre del 2001

El circuito permite a un microcontrolador BasicStampII convertir el valor de hasta 15 potenciómetros en información MIDI y permite modificar el programa del microcontrolador por medio del puerto DB9. El circuito puede ser alimentado con una pila de 9V e incluye un interruptor para prender y apagar y un led para ver si está prendido o apagado. El circuito incluye 8 potenciómetros y 7 jacks para pedales u otros tipos de controladores.

Figura 15.14. | Diagrama del circuito GAB.

Para evitar falsos contactos y posibles cortos se recomienda diseñar un pequeño circuito en la placa de cobre. Existen varios procesos caseros que permiten el diseño de esta clase de circuitos.

La figura 15.15 muestra en acetato (para facilitar el copiado) los diagramas en tamaño real para las caras superior e inferior para la placa de cobre de 10×10 cm.

EXPLICACIÓN Y TRANSCRIPCIÓN DEL CÓDIGO FUENTE

En la primera parte del programa se definen las constantes requeridas. Es importante recalcar que el valor 32780 de la constante midibaudmode corresponde al valor que el BasicStamp necesita para realizar la transmisión a la velocidad y con las características que el protocolo

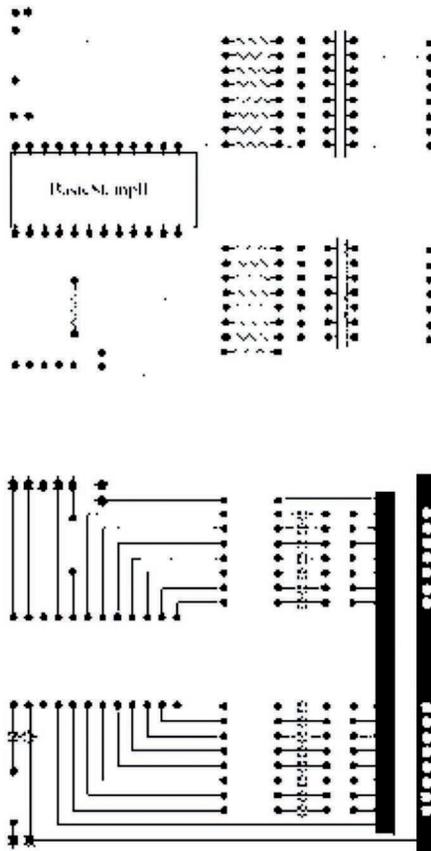


Figura 15.15. Diagrama del circuito GAB.

MIDI determina: 31.25(+/-1%) Kbaud, asincrónico, con un bit de inicio, ocho bits de datos y un bit final. El valor binario de la constante controlador %10110000 corresponde al *status byte* asignado al dato MIDI ControlChange canal 1. Después de definir las constantes se definen las variables necesarias y se declaran como salida todos los puertos del BasicStamp para evitar daños en el circuito. La rutina principal se encarga de monitorear constantemente el valor de los potenciómetros por medio de la instrucción RCTime. Si el valor de alguno de ellos cambia, la rutina *mandaDato* se encarga de construir el dato MIDI correspondiente y transmitirlo a la velocidad requerida.

A continuación se transcribe íntegro y sin modificaciones el código fuente del programa:

```

1. ' GAB controlador MIDI Versión 1.0 Autor Hugo
2. ' Solís Fecha 25 de octubre del 2001 Este programa
3. ' hace posible que el BasicStampII lea hasta 15
4. ' potenciómetros y convierta sus valores en valores
5. ' MIDI ControlChange 16 - 23 canal 1 el programa
6. ' esta diseñado para ser utilizado en el
7. ' BasicStamp del GAB controlador MIDI. Para
8. ' detalles de funcionamiento ver la tesis «Gab
9. ' 1.0, sistema de reinterpretación electrónica
10. ' para pianos acústicos» de Hugo Solís
11.
12.
13. ' Declaración de constantes
14. '31.25kb, 8n1, non-inverted, open collector
15. midibaudmode con 32780
16. 'numero binario correspondiente al
17. 'status byte Control Change canal 1
18. controlador con %10110000
19. pinDeSalida con 0
20. defase con 15
21. cantidadDeControladores con 8
22.
23. ' Declaración de variables
24. resultado var word
25. valor var byte
26. valorAnterior var byte(cantidadDeControladores)
27. numeroDePin var nib
28.
29. ' Para seguridad del Microcontrolador,
30. ' todos los pines no utilizados
31. ' deben de declararse como puertos de salida
32. DIRS = %1111111111111111 'all outs
33. OUTS = %0000000000000000 'all low
34.
35.
36. principal:
37. for numeroDePin = 1 to cantidadDeControladores
38.   high numeroDePin
39.   pause 1
40.   RCTIME numeroDePin, 1, resultado
41.   valor = ((resultado Max 414)* 127 / 414)
42.   if valor = valorAnterior(numeroDePin - 1)
43.     then noMandaDato
44.     gosub mandaDato
45.   noMandaDato:
46.     valorAnterior(numeroDePin - 1) = valor
47.   next
48. goto principal
49.
50.
51. mandaDato

```

```
52. ´ quitar la coma inicial de la linea para
53. ´ ver valores en pantalla
54. ´ debug cls, dec numeroDePin, dec valor Max 127 <
55. serout pinDeSalida, midibaudmode,
56. [controlador, numeroDePin + defase, valor]
57. return
58.
```

CONCLUSIONES

El sistema GAB es una herramienta interpretativa, planeada y diseñada por un músico. Si bien es cierto que para su creación fue necesario un trabajo interdisciplinario que involucró aspectos de computación y electrónica, la preocupación musical siempre estuvo presente. De nada serviría todo el trabajo realizado si el resultado musical no fuese interesante. Lo valioso que pueda llegar a ser este proyecto dependerá únicamente de la forma en que sea utilizado. La herramienta está hecha, viene ahora el proceso de trabajar con ella, de aprovechar sus recursos y conocer sus limitaciones; también de las limitaciones se puede sacar provecho. Las siguientes versiones del sistema GAB y las mejoras que se le vayan haciendo serán únicamente el resultado de trabajar con esta primera versión.

Vivimos una época en donde la tecnología se ha hecho presente en todos los ámbitos del quehacer humano: el arte y la música no son la excepción. Todos estamos en mayor o menor medida, conscientes o inconscientes, inmersos y relacionados con el mundo de las computadoras y los circuitos electrónicos. Sin embargo, mientras más evoluciona la tecnología más difícil es entender cómo funciona. El sistema GAB es un pretexto y una invitación a no hacer de los aparatos y programas musicales “cajas negras” que compramos en dólares y que con sus particulares recursos definen incluso nuestros proyectos musicales. El sistema GAB es sin duda mucho más limitado que los equipos comerciales; sin embargo, es una opción para ciertos procesos creativos. ■ ■ ■

Interface de música y pintura improvisadora

Este proyecto fue realizado como parte de la maestría en Arte y Ciencia en el Media Laboratory del MIT.

MOTIVACIÓN

La idea de crear un sistema audiovisual que pudiera relacionar la improvisación musical con instrumentos acústicos y gráficos digitales dinámicos fue un viejo deseo que se gestó durante mi formación como pianista y, sobre todo, que sumaba mi experiencia al trabajar con instrumentos acústicos en la ejecución, porque eso fue un eje importante para crear mi visión musical y porque desde muy temprano me interesé por el campo de la composición y de la improvisación musical; además, un eslabón de mis estudios se centró en esa dirección. Una de las actividades más interesantes que desarrollé como parte del programa de composición musical en el Laboratorio de Creación Musical de la Escuela Nacional de Música de México fue la realización de sesiones imaginarias. Durante estas sesiones de dos horas los estudiantes cerrábamos los ojos y describíamos verbalmente nuestras fantasías creativas. En la mayoría de los casos, las descripciones iniciales eran vagas y difusas, pero tratando de influir lo menos posible el profesor intentaba ayudar a cada estudiante para que aclarara y detallara sus ideas, haciendo preguntas sobre los eventos que los estudiantes describíamos durante

nuestro proceso. En estas sesiones las ideas se iban haciendo más claras y precisas gradualmente y, al final de cada experiencia, una fantasía corta, pero precisa, se podía recuperar y era recreada por los estudiantes para aterrizarla en un material concreto y detonador de la experiencia creadora.

La combinación de la ejecución de piano y de los ejercicios de composición musical me llevaron al campo de la improvisación durante muchos años, pero centré mi atención en la producción de improvisación acústica. Un poco más tarde me atrajo el uso de la electrónica con la idea de que nuevos sonidos y nuevas tecnologías ampliaran mi experiencia como improvisador y, por tanto, comencé a improvisar con dispositivos electrónicos (principalmente sintetizadores comerciales y software de computadora). Al tiempo que desarrollaba un interés en la improvisación empecé a tener una profunda inclinación por la expresión visual (más como espectador que como creador); sin embargo, mi interés en las posibilidades expresivas de mezclar música y pintura me llevó a crear el Trío Nicrom en el año 2001 (ver *ch:musica*). El Trío Nicrom fue un ensamble compuesto por un pintor, un instrumento de aliento (clarinete) y yo mismo como músico con algún dispositivo electrónico. El material audiovisual era completamente improvisado en el escenario mezclando los sonidos electroacústicos y el material visual creado para el instante. Las pinturas creadas en tiempo real mostradas en el escenario con la ayuda de un proyector, así como las relaciones entre el sonido y la imagen eran completamente subjetivas, pero destacaba la interacción real entre los artistas porque, aunque no era determinista, el resultado audiovisual resultaba (en la mayoría de los casos) un producto orgánico y unificado.

Cuando entré en el Laboratorio de Medios del MIT, me di cuenta de que para entonces ya se había hecho una gran cantidad de trabajos en el campo del desempeño audiovisual y que podría estudiar algunas de las obras de la disciplina e interactuar con personas preocupadas por el campo. Al mismo tiempo podría familiarizarme con las tecnologías asociadas con este tipo de expresión. Sobre todo, me di cuenta de que el proceso creativo podía extenderse más allá del uso de la tecnología disponible y llegar al punto en el que la creación de las herramientas se convirtiera en parte del desarrollo artístico.

En este contexto era natural planificar y desarrollar un sistema donde mis intereses – desempeño musical, improvisación colectiva y creación audiovisual– se fusionaran y así

fue cómo de manera natural dentro de mi proceso de crecimiento realicé la Interface de música y pintura improvisatoria (IMPI) durante el segundo semestre del año 2003 y los primeros seis meses de 2004.

DESCRIPCIÓN

La interface de música y pintura improvisatoria (IMPI) es un software para la creación y orientación de improvisaciones audiovisuales que opera al dibujar en una tableta digital, utilizando una sintaxis definida y un conductor que genera (en tiempo real) tanto la música que se muestra en varios formatos de pantallas de ordenador para los músicos, así como los gráficos dinámicos proyectados en el escenario en sincronía con la música que se escucha y que ha sido interpretada por los ejecutantes. El tipo de estilo musical al que se dirige este sistema es no tonal, no rítmico y orientado a la textura, esto significa que se hace énfasis en el control de las cualidades tímbricas y en las transiciones continuas. Uno de los principales objetivos del sistema consiste en la traducción de elementos de composición planificados, así como en la estructura precisa y en la sincronía entre instrumentos con dominio de la improvisación. Los gráficos que IMPI genera son orgánicos, fluidos, vivos, dinámicos y unificados con la música. La mayoría de los sistemas audiovisuales actualmente desarrollados y la mayor parte de la investigación en el campo se centran en la exploración y en la creación de sistemas audiovisuales que tienen su salida de audio con sonidos electrónicos, en cambio, este trabajo, propone la creación de un sistema audiovisual donde la música se genera por instrumentos acústicos en tiempo real.

IMPI es un proyecto en desarrollo. Actualmente se han creado tres versiones donde cada una resuelve diferentes desafíos técnicos y estéticos. Lo que se ha desarrollado hasta ahora es un paso importante en la creación de una versión final que integre todas las características y que produzca el resultado que al que está destinado el sistema: crear una improvisación colectiva libre para obtener gestos musicales con acontecimientos sincronizados. Se sabe que no es una tarea fácil, por tanto, este proyecto, propone un sistema audiovisual para la obtención de resultados audibles concretos creados por músicos en tiempo real.

Dado que no existe un método único y objetivo para correlacionar elementos acústicos y visuales, y aunque ya se han creado varias obras en el ámbito de la interpretación audiovisual con su propio paradigma y técnicas, IMPI pretende ser una contribución al campo.

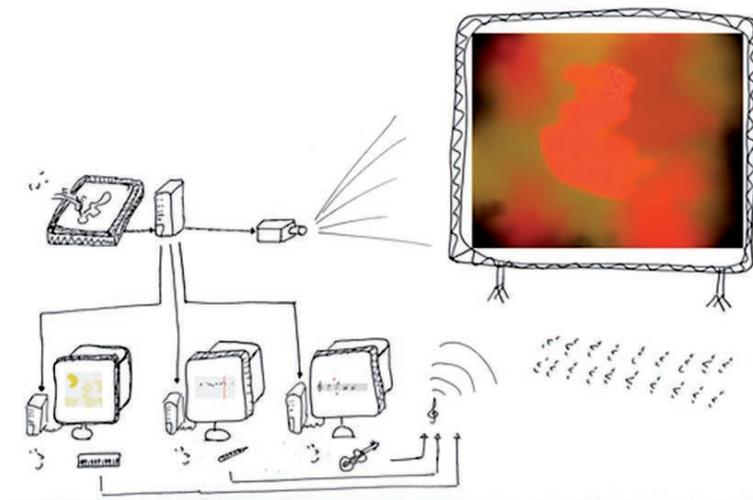


Figura 16.1. | Diagrama funcional de IMPI.

IMPI es un sistema audiovisual que tiene dos objetivos principales: *i*) ayudar a conjuntos de músicos acústicos en la creación de improvisaciones orgánicas en las que puedan crearse elementos de la técnica compositiva, que incluya la sincronización entre instrumentos, así como cambios abruptos del material sonoro; *ii*) crear una expresión gráfica dinámica que, junto con la música, produzca una obra audiovisual coherente y maleable con significado perceptual entre todos los elementos.

En las versiones que se han desarrollado, un “conductor” se encarga de dar forma a las improvisaciones y crear el material visual dibujando con una sintaxis específica un referente en una tableta digital. Con el objetivo de reconocer el gesto del recorrido de la pieza se analizan varios elementos como velocidad, presión, ubicación y duración del trazo, y en las versiones desarrolladas, los elementos musicales que el “conductor” puede controlar, el nivel de participación y de actividad de cada miembro del conjunto sumándose referencias tales como el contorno general de la improvisación, los niveles de intensidad y las regiones tona-

les y modulaciones de la música. Los elementos visuales que pueden ser controlados por el conductor son la forma, la ubicación y el color de los objetos. El resto de las funciones se automatiza utilizando las funciones que se explican más adelante en la sección “Versiones y proceso de evolución”.

La información musical generada por el trazo del conductor es distribuida y enviada al conjunto por una red informática y para su operación —idealmente— cada músico debe tener una computadora para leer, en sustitución de la partitura de papel tradicional. En la pantalla cada músico recibe el material que el conductor está creando para él o ella y, aunque en este punto se presenta una especie de partitura abierta, porque la información se representa con varios grados de abstracción, los músicos son libres de elegir qué representación seguir. La notación de música estándar (SMN) es una de las posibles representaciones y debe estar a la vista mientras se desplaza por la ventana, ya que las formas dinámicas que representan el contenido musical son otro tipo de simbología. En este caso, el pintor-conductor tiene el potencial para una mayor libertad; sin embargo, se conserva la idea de sostener una improvisación libre.

Como dije anteriormente, la información visual también es generada por los dibujos del conductor, ya que los gestos utilizados en los dibujos (trazos) producen imágenes dinámicas que deben proyectarse en el escenario con la música. En las versiones implementadas de IMPI, esos trazos creados por el conductor se asignan directamente a la forma del objeto creado en la salida visual principal representada en la pantalla. Los filtros dinámicos y las transformaciones se aplicarán posteriormente a la imagen para generar una fusión entre las formas y unas alteraciones determinadas que promuevan el desarrollo formal de la pieza musical.

El tipo de música que el sistema genera no está basado en el sistema tonal ni rítmico y no está orientado a patrones, esto último significa que se debe poner énfasis en el control de las cualidades tímbricas y las texturas de producción del sonido. Debido a que la interface fue creada para instrumentos acústicos hay un interés particular en el elemento del timbre y por ello los instrumentos deben utilizar todos sus recursos de técnicas extendidas de ejecución y con estos recursos representar la intención y crear la sensación de transformaciones con-

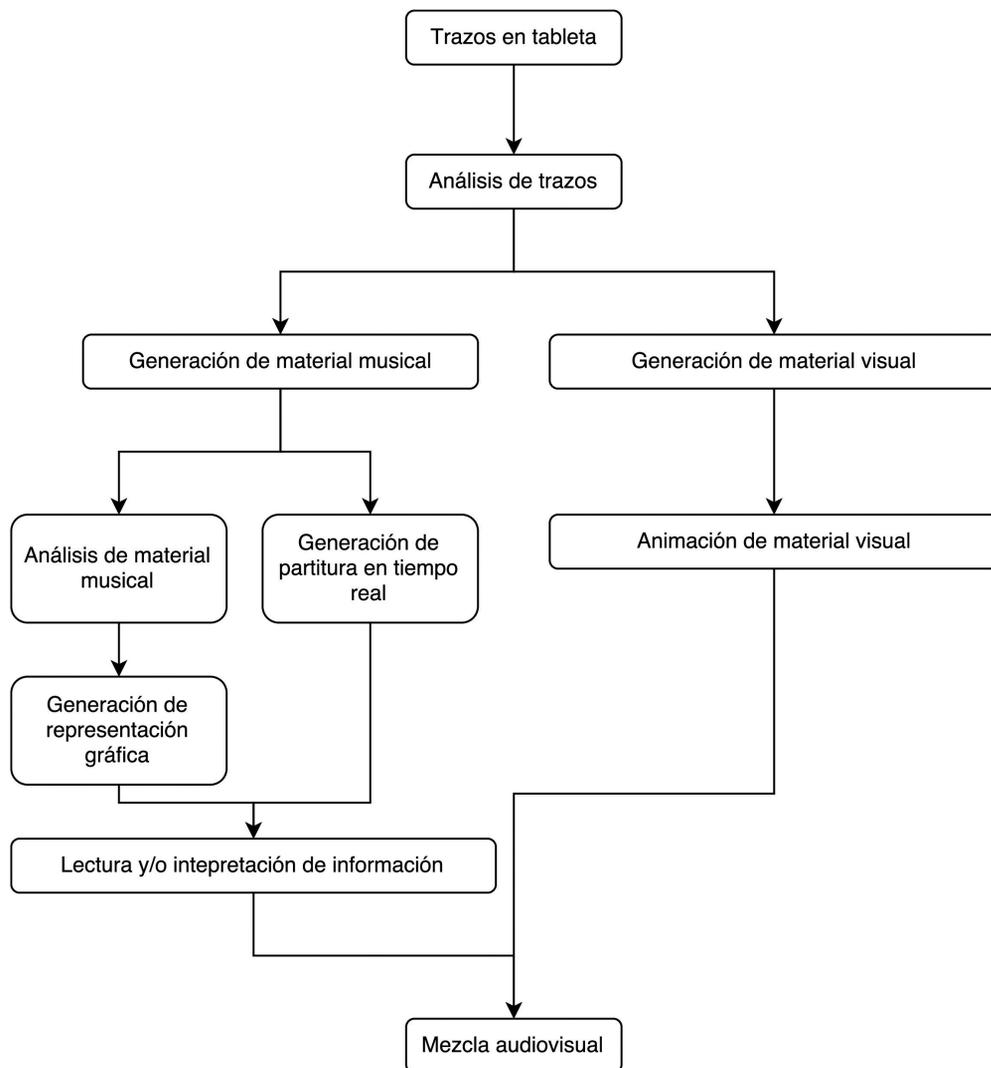


Figura 16.2. | Diagrama de flujo del sistema IMPI.

tinuas en el tiempo. En relación con el aspecto tonal, la sugerencia es recurrir a intervalos más pequeños que el semitono y, si el instrumento lo permite, utilizar glissandi.

Con el fin de crear la sensación de movimiento sobre el espacio en el dominio musical, el sistema IMPI debe utilizar transformaciones fundamentales en el tono y en la intensidad y, para crear la sensación de transformación continua en el dominio del tiempo, la sincro-

nía de los acontecimientos y el uso de pulsos regulares o patrones rítmicos claros deben ser empleados con cautela, evitando politempis, patrones rítmicos irregulares y repeticiones. En la representación del sistema IMPI, los gráficos (en el sistema) deben ser extremadamente maleables y orgánicos, las formas deben cambiar constantemente de dimensión y contorno vívida y evidente. Los detalles de este ideal gráfico se describen en la siguiente sección.

ESTÉTICA DEL SISTEMA

IMPI es un sistema que configura y materializa fantasías cinéticas con ciertas características. Los elementos centrales que integran estas ilusiones *arquetípicas* más allá de los medios para representarlas son:

Libre movimiento de objetos Tomando como referencia la ilusión de que los objetos que forman parte de la representación pueden volar y moverse por el espacio en todas las direcciones, una buena analogía con el mundo natural sería el vuelo de una mosca.

Nubes de objetos. Me imagino colecciones de objetos que forman grandes nubes que se pueden expandir y colapsar con el tiempo moviéndose orgánicamente, esta representación es similar a la formas que crean las parvadas de aves.

Transformaciones graduales Me interesan las transformaciones lentas pero constantes del material, esos continuos y sutiles cambios de un punto a otro aparentemente imperceptibles que en el ámbito más amplio crean modificaciones. Muchos eventos físicos y biológicos tienen esta calidad en su forma de evolucionar. Para mí, algunos de los sistemas dinámicos más interesantes son el movimiento de fluidos, del humo y de las nubes.

Forma y textura orgánica No importa si se trata de un evento acústico o visual, su forma y textura deben ser granulares o fibrosos.

Los elementos descritos anteriormente deben materializarse necesariamente en elementos musicales físicos y medibles. No hay una traducción objetiva de estas ilusiones al mundo musical gráfico, y las técnicas y procedimientos para crear tal traducción son realmente parte del proceso creativo. IMPI tiene su propio paradigma en la forma en que crea la traducción de estas ideas en materiales acústicos y visuales.

TECNOLOGÍA

Java

IMPI se desarrolló utilizando el lenguaje Java 1.4. Java es un lenguaje agradable que ofrece un equilibrio entre lenguajes de bajo nivel y de alto nivel [Sun04] y es mucho más sencillo que C y C++. Además de lo anterior, Java es (al menos teóricamente) un lenguaje multiplataforma, lo que significa que el código escrito en Java puro puede ejecutarse en cualquier plataforma. La idea de ejecutar el mismo código en un sistema Windows, Macintosh, Linux o PDA sin tener que cambiar el código es una característica muy atractiva que concibe inclusión desde la concepción del proyecto. Otra buena característica de Java es el gran número de bibliotecas que se han creado para ello. La gente ha creado paquetes específicos para una gran cantidad de tareas, incluyendo audio MIDI, gráficos 2D, gráficos 3D y manipulaciones de video.

Java requiere una máquina virtual que funciona como un puente entre el sistema operativo y el código, y eso lo hace lento para el procesamiento multimedia. Para hacer frente a esta situación recurrí a la creación de bibliotecas en C que fueran utilizadas por Java. Los protocolos de red, el control de hardware y la comunicación con otros idiomas fueron otras de las características que convirtieron a Java en el lenguaje idóneo para el proyecto.

Jsyn, por ejemplo, utiliza un motor de audio que es una biblioteca C para manejar el hardware de audio. El núcleo de audio está oculto del usuario; sin embargo, destaca que es posible llamarlo desde Java. Siempre y cuando la biblioteca esté presente en el sistema, el código sigue siendo multiplataforma.

Con el aumento de la velocidad de las computadoras, el problema del rendimiento de Java se ha resuelto de manera natural, lo que no era posible hace dos o tres años. Ahora se puede ejecutar fácilmente en una computadora de características regulares. IMPI utiliza una representación gráfica compleja para la parte visual de la interface; los filtros y los algoritmos que crean la animación consumen casi toda la CPU de la potente computadora actual. Sin embargo, IMPI es un sistema que utiliza una computadora para cada ejecutante, esto significa que al menos la representación de la partitura y los gráficos de cada ejecutante se calculan en una CPU por separado.

JSML

El lenguaje de la especificación de la música de Java (JMSL) fue creado por Nick Didkovsky y Phil Burk, como sucesor evolutivo del lenguaje de la especificación de la música jerárquica (HMSL) que había sido creado por Phil Burk, Larry Polansky y David Rosenboom. JMSL es una “herramienta de desarrollo basada en Java para experimentos en composición algorítmica, ejecución en vivo y diseño de instrumentos inteligentes” (Didkovsky y Burk, 2001). JMSL es un excelente paquete para la creación de música algorítmica, sus conceptos de organización jerárquica, las tareas musicales y las colecciones paralelas y secuenciales resultan totalmente naturales para la organización de la música, sin importar qué tipo de estructuras y elementos se crean en el lenguaje musical.

Además, JMSL está bien integrado con JSyn porque utiliza métodos nativos escritos en C para proporcionar síntesis de audio en tiempo real para programadores Java. JSyn se puede utilizar para generar efectos de sonido, entornos de audio o música y se basa en el modelo tradicional de generadores de unidades que pueden ser conectados entre sí para formar sonidos complejos (Didkovsky y Burk, 2001). Esto significa que si en el futuro IMPI crece para integrar la producción y la manipulación de música electrónica o electroacústica la ampliación sería fácil de integrar, ya que la versión actual de JMSL (101) ofrece la posibilidad de renderizar puntajes en tiempo real.

La primera versión de IMPI utilizó su propia representación de puntuación, esto se muestra en la sección “Versiones y el proceso de evolución”; durante el proceso de mejorar el sistema, la opción de usar el paquete de puntuación de JMSL parecía mucho más lógica porque el paquete de partituras de JMSL es un paquete de Java totalmente integrado con el resto de JMSL y porque permite la creación y transformación de partituras escritas con notación musical estándar. El método para crear, modificar o transformar la partitura está totalmente abierto a la elección del programador y puede hacerse en tiempo real en su totalidad. Además de lo descrito anteriormente, las partituras del JMSL se pueden guardar como partituras musicXML y esto significa que el archivo final se puede abrir en editores de música como Finale, Turandot e Igor Engraver.

HARDWARE

IMPI utiliza una red informática para transmitir la información entre los músicos. En estos días, las redes de ordenadores son fáciles de configurar, ya que los protocolos se han extendido y el hardware necesario no es tan caro como solía ser en su fase temprana de implementación. Además, la opción de crear redes inalámbricas rápidas parece ser una buena solución para configurar sistemas de ejecución en vivo.

La configuración ideal para el uso del sistema IMPI es una red inalámbrica de computadoras portátiles donde cada músico tiene su propia computadora y donde todas las computadoras deben ser lo suficientemente rápidas para renderizar las animaciones gráficas y la puntuación y, además de la velocidad, deben tener una conexión LAN inalámbrica. El conductor también necesita una computadora con una tableta digital conectada a la salida de video del proyector que muestra los gráficos principales. Para la transmisión en red se requiere un enrutador inalámbrico, en su momento se utilizó el protocolo de comunicaciones 802.11b.

Tableta digital

IMPI utiliza una tableta digital 12 × 12 Intuos2 de Wacom (<https://www.wacom.com/>). Este dispositivo registra todos los movimientos de un lápiz especial que se dibujan en la superficie de la tableta. Los parámetros que se capturan con una velocidad máxima de 200 puntos por segundo son:

Ubicación La tableta registra las coordenadas x, y de la pluma con una resolución de 2 540 puntos por pulgada.

Presión La tableta registra la presión aplicada con la pluma en la superficie de la tableta con una resolución de 1 024 niveles.

Ángulo La tableta registra la dirección en la que el usuario sostiene el lápiz.

Inclinación La tableta detecta la inclinación en la que el usuario está sujetando la pluma.

Botones y dirección empleados El dispositivo registra si se presiona cualquiera de los tres botones de la pluma. También registra qué extremo de la pluma se está utilizando.

Aprovechando los muchos parámetros que se pueden detectar con este dispositivo, IMPI recurre a algunos de ellos para crear un instrumento audiovisual con varias características, pero la forma en que se han utilizado estos parámetros varía entre las diferentes versiones del sistema; algunas de ellas se han conservado y esas que se mantienen son las estrategias centrales de la interface.

Ubicación IMPI rastrea la ubicación del lápiz para registrar la forma del trazo (golpe) creado por el dibujante-conductor. Los valores se ajustan a los valores normales empleados con un mouse.

Velocidad Medir la ubicación y el tiempo permite obtener la velocidad de trazo en cada momento. La velocidad del gesto es uno de los elementos más importantes para entender el tipo de trazo que el conductor está realizando. El rango se escala como un número decimal entre 0.0 y 1.0. La velocidad más rápida se calculó sobre la base de la velocidad media más rápida que podría ser significativa para el rendimiento del sistema. Si se crea un trazo con una velocidad más rápida se aplica un valor límite.

Presión La presión de la pluma también contiene información significativa que ayuda a entender el tipo de gesto que el pintor-conductor está creando. El valor se escala como un número decimal entre 0.0 y 1.0.

Posición El ángulo y la inclinación de la pluma se combinan en un solo valor que representa hasta qué punto la parte delantera o trasera de la tableta está en el extremo exterior de la pluma en relación con el extremo de la pluma que está tocando la tableta. Imagínese examinar la inclinación a lo largo del eje y en un *joystick* sin tener en cuenta el eje x. La decisión de fusionar ambos valores fue considerada con la idea de que el movimiento de retroceso es un movimiento natural con un significado intuitivo. La fusión se creó multiplicando ambos valores. El valor también se escaló como un número decimal entre 0.0 y 1.0.

VERSIONES Y PROCESO DE EVOLUCIÓN

La creación de IMPI ha sido un proceso largo. Se planificaron, esbozaron, diseñaron e implementaron varios prototipos y versiones y algunas de estas versiones fueron ideas que nunca tuvieron su salida de código porque las ideas y los sistemas eran demasiado vagos o demasiado ambiciosos, o porque no tenía sentido desarrollar más versiones sin optimizar las que estaba trabajando. Así fue como se implementaron tres versiones que se desarrollaron a profundidad. Dado que la evolución de IMPI progresó paralelamente con mi aprendizaje en las tecnologías empleadas y en las ideas audiovisuales, durante su creación tuve que crear tecnologías que más tarde tuve que descartar porque el desarrollo de IMPI no se comportaba como una línea recta de progreso constante, sino como una exploración con altibajos, mejor descrito como un proceso de aprendizaje que llevaba una trayectoria en espiral. Esta decantación también ocurrió con los recursos: me pasó con algunas bibliotecas y con algunos paquetes de otros creadores que previamente fueron importados e implementados en alguna etapa del desarrollo, algunos de los cuales fueron posteriormente descartados o cambiados por otras soluciones más óptimas.

IMPI es un sistema perfectible y aún se puede hacer mucho trabajo para mejorarlo; sin embargo, la versión actual es la más estéticamente equilibrada entre todas las implementadas con anterioridad.

Es importante notar que aunque hay cambios en la forma en que los intérpretes interactúan con el sistema y aunque hay cambios en las tecnologías usadas entre las versiones, hay una estética constante que se manifiesta y sostiene durante todo el proceso. El tipo de intención acústica y la esencia del material visual han sido los mismos en todas las versiones. Una explicación detallada de esta búsqueda estética se describe al comienzo de este capítulo.

IDEA ORIGINAL

La idea original era reemplazar la figura del director tradicional de música de cámara con un músico-pintor que pudiera guiar el espectáculo con su pintura. Para estos conciertos los músicos tendrían un rango estrecho para modificar el material acústico porque recibirían la

partitura en la pantalla de sus computadoras, ya que los dibujos que el conductor crea se utilizan para definir parámetros musicales en las partituras en tiempo real.

En esta idea inicial ni los gráficos ni las relaciones entre música e imagen estaban bien definidos, sólo se mantuvo la intención de desarrollar un marco que pudiera crear los materiales audiovisuales siguiendo la estética que detonó la idea inicial. La primera descripción escrita del sistema, detallada en la propuesta de tesis, establece que

el lenguaje gráfico permite el control jerárquico del material musical. En primer lugar, los gráficos iniciales (formas geométricas) controlan el nivel microscópico musical como el tono, la duración, el timbre y la intensidad de una nota. Luego, las acciones sobre los objetos gráficos (disolución, movimiento, rotación) permiten la modificación de elementos musicales tales como contornos rítmicos, melódicos y dinámicos de frases y, finalmente, la modificación general de los gráficos (congelación, rotación, difuminación) controla los elementos macro de la música, tales como rango dinámico, rango de tono y clímax.

PRIMERA IMPLEMENTACIÓN

Sobre la base de la idea inicial y como una exploración de las tecnologías que se utilizarían se creó la primera aplicación. En ese momento, el centro de la investigación estaba más centrado en resolver los desafíos computacionales que en el resultado estético. Aunque utilizable, la tableta digital no se implementó en esta versión y todo el código fue desarrollado para el uso de un ratón.

En esta primera versión los músicos sólo tienen una ventana en sus pantallas con la puntuación y el pintor tiene una ventana donde puede dibujar líneas con el ratón, así como una

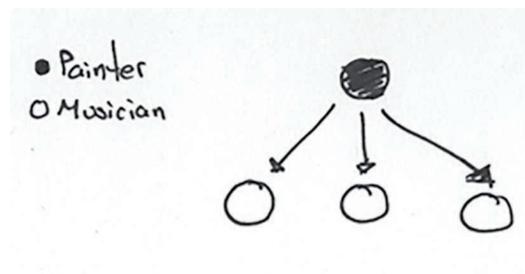


Figura 16.3. | Estructura tradicional de dirección de ensambles.

pequeña ventana con una paleta de diferentes colores donde puede elegir el color con el clic del ratón.

A pesar de que técnicamente es posible tener múltiples pintores y múltiples músicos en la red, esta versión fue concebida como un sistema en el que sólo un pintor crea el material para múltiples músicos (figura 16.3) porque, en la práctica, esta versión envía la misma información a todos los músicos y las relaciones audiovisuales no fueron planeadas artísticamente con lo cual esto no resulta útil en situaciones musicales reales.

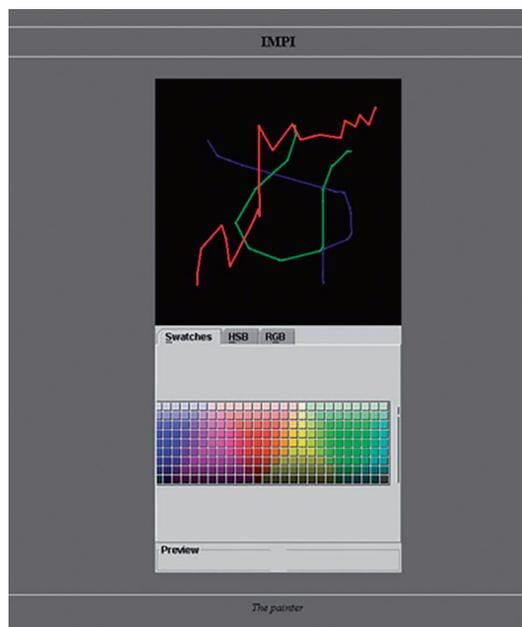


Figura 16.4. | Lienzo para el trazo.

El pintor

El lado del pintor del sistema consiste en una paleta de colores y un lienzo negro (figura 16.4). El pintor-conductor utiliza el ratón para dibujar líneas en el color seleccionado donde no hay tipos de asignaciones, filtros o transformaciones de la entrada y la pantalla muestra el movimiento exacto del ratón. Las líneas que el usuario crea desaparecen gradualmente con el tiempo al volverse cada vez más oscuras. No hay relación entre este proceso de ani-

mación y la música creada, y los elementos para crear la parte musical toman como referencia desde el momento en que se genera el gesto.

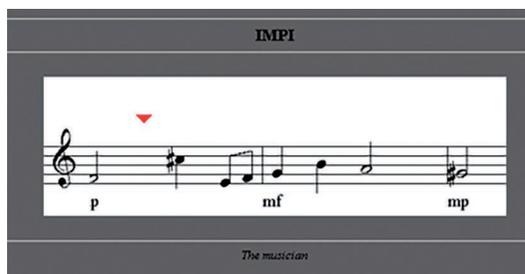


Figura 16.5. | Partitura dinámica.

Los músicos

Los músicos tienen en su pantalla sólo una ventana donde se representa la partitura (figura 16.5). El papel de la computadora en relación con el músico se limita a proporcionar un tempo que se puede crear en tiempo real.

Relación audiovisual

El mapeo entre lo musical y lo visual en esta versión no fue profundamente explorado; sin embargo, algunos elementos interesantes que surgieron de esta versión se tuvieron en cuenta para las versiones posteriores. Se consideraron los siguientes parámetros para generar la música cuando el usuario dibuja con el ratón una línea en el lienzo:

Color El color actual de la línea se utiliza para determinar la intensidad de la frase musical. Los valores rojo, verde y azul se suman y el resultado se escala en un rango entre uno y cuatro. Si el valor es uno (obtenido con colores oscuros) la intensidad de la frase es mezzoforte, mientras que el color se vuelve más brillante; el rango de dinámica posible, el cual es elegido al azar, también se incrementa. Si el valor del color es cuatro (color blanco) la dinámica de la frase musical podría ser un crescendo desde pp, p, mp, mf, f, ff a fff.

Longitud La longitud se obtiene midiendo y añadiendo la distancia entre todos los puntos de la línea. La longitud de la línea se utiliza para determinar el número de notas que tendrá cada medida dentro de la frase musical. Los valores razonables se establecen como el valor mínimo y máximo para la longitud. El valor se escala en un rango entre uno y cuatro. Si el valor es uno (líneas cortas) las medidas se crean con tres notas y a medida que la longitud del movimiento se hace más grande, el rango del número posible de notas, de las cuales el número final de notas se elige al azar, también se incrementa, pasando de uno a cuatro. Si el valor es cuatro, la medida podría tener entre una nota (nota blanca o resto) y cuatro notas de valores diferentes. La longitud de la línea sólo establece el número de notas por medida, no la duración de las mismas.

Duración El tiempo que se tarda en crear una línea se utiliza para establecer el tipo de patrón rítmico. Los valores razonables se establecen como valores mínimo y máximo para la duración. Los limitadores se aplican si la duración está fuera del rango razonable. El valor se escala en un rango entre uno y cuatro; si el valor es uno la medida tiene sólo un tipo de patrón rítmico, teniendo en cuenta el número de notas, pero a medida que la duración en la creación de la línea se hace más grande, los posibles patrones rítmicos elegidos al azar también se incrementan. Si el número de notas en la medida es uno, el valor de duración no tendrá ningún efecto, ya que sólo hay una única forma de llenar la medida con una nota. Por otro lado, si el número de notas en la medida es cuatro una larga duración permitiría muchos patrones diferentes. Los posibles patrones se toman de la tabla descrita en la sección Render de puntuación.

Curvatura Se utiliza un algoritmo básico para medir la curvatura de la línea para establecer los tipos de tono para la frase musical. Se miden todos los ángulos creados con los segmentos formados entre puntos adyacentes (figura 16.6). Si el ángulo está cerca de 180° , esta sección de la línea es casi recta y se agrega una etiqueta de valor mínimo a un contador; si el ángulo es alrededor de 90° se agrega una etiqueta de valor medio y si el ángulo es cercano a 0° significa que justo en esa sección de la línea hay un gran cambio de dirección y se agrega un valor alto al contador. El valor final del contador arroja una idea sobre qué tan ondulada es la línea. El valor se escala entre uno y cuatro; si el valor es uno todas las notas de la frase son c una octava por encima de la

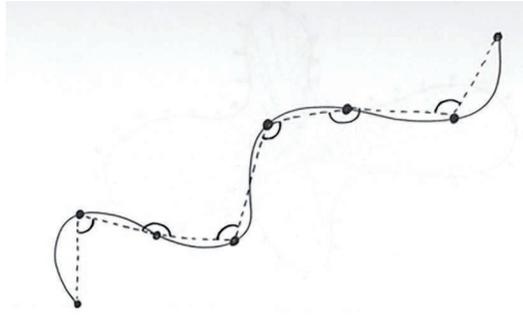


Figura 16.6. | Curvatura de curva y puntos de medición.

mitad de c y, mientras que el valor aumente, el rango de notas posibles para ser elegido al azar también se incrementará. Para los propósitos de la prueba se fijó una gama para que cupiera dentro del pentagrama, sin líneas adicionales.

Velocidad Con la duración total de la carrera y con la longitud final de la línea se puede calcular la velocidad media a la que fue creado el trazo. Este valor se utiliza para determinar el *tempo* de la música. La gama comprende de 60 a 240 BPM donde el valor del tempo se refleja en la velocidad a la que la partitura se desplaza en la pantalla.

El generador de partituras

La principal contribución de esta versión consistió en el desarrollo de la biblioteca de renderización de puntuación en tiempo real. Utilizando las capacidades gráficas de Java 2D y las fuentes Toccata y Fughetta (<http://members.efn.org/~bch/fullpackage.html>) se implementó una partitura que se desplaza constantemente de derecha a izquierda sobre la pantalla. En la esquina superior derecha de la clave un triángulo muestra la posición actual dentro de la partitura.

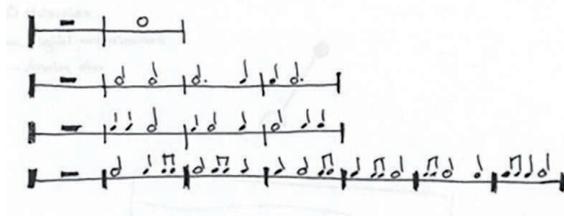


Figura 16.7. | Posibles ritmos de la primera versión de IMPI.

Una tabla de números enteros define las posibles fórmulas rítmicas que la partitura puede representar (figura 16.7). Los patrones básicos se seleccionaron para fines de prueba, pero podrían extenderse para implementar otras células rítmicas de mayor complejidad.

Observaciones de la primera versión

En el campo técnico la principal contribución de esta versión fue la implementación de la renderización de la puntuación y la implementación de la etapa inicial del motor de red. Esta versión me ayudó a entender muchas de las funciones gráficas de Java para el dominio estético, sobre todo, a observar que la mayoría del mapeo directo entre el campo visual y el gráfico era demasiado directo para tener un valor significativo. La experiencia en el desarrollo de esta versión contribuyó a la implementación de una siguiente versión.

SEGUNDA IMPLEMENTACIÓN

Basándome en la experiencia de haber diseñado la primera versión y después de considerar seriamente cuál sería la mejor dirección para la evolución del sistema trabajé en una segunda versión que mantuviera varias similitudes con la primera, pero añadiendo cambios significativos. Parte del código que se empleó en la primera versión fue tomada como punto de partida y dado que esta versión estuvo más elaborada que la anterior (tanto técnica como musicalmente) fue utilizada durante un taller con John Zorn la cual se describe más adelante.

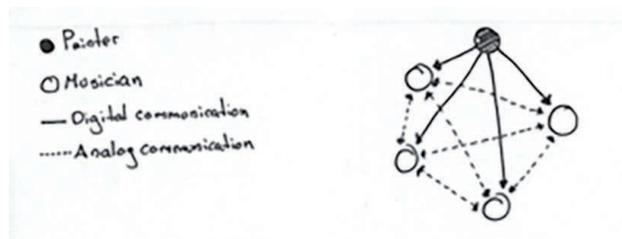


Figura 16.8. | Configuración de ensamblaje para control de IMPI.

Configuración del ensamble

Esta versión conserva la configuración de tener un único pintor que dirige a un grupo de músicos (figura 16.8) donde éstos pueden recibir datos, pero no tienen la capacidad técnica de comunicarse con el resto del conjunto. Pero aún así tienen la opción de crear música sin (necesariamente) seguir la partitura, como se explicará más adelante, en “Los ejecutantes”.

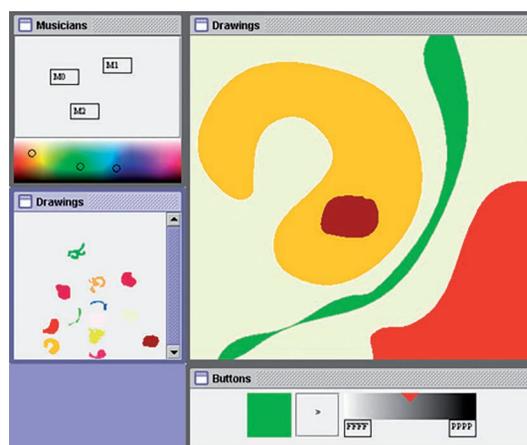


Figura 16.9. | Ventana del director en la versión dos de IMPI.

El pintor

En esta versión el conductor puede utilizar varias características para guiar la improvisación. La ventana del conductor (figura 16.9) está dividida en cuatro secciones donde cada una tiene una función particular en el sistema.

La ventana de dibujo

La ventana principal es el lienzo (figura 16.10). En éste el conductor puede dibujar las formas que generan los objetos musicales y puede ver el resultado gráfico que se proyecta para el público. Cuando la ejecución da inicio, el lienzo es un área blanca rectangular. Contrariamente a la primera versión, ésta recurre a las formas como la unidad gráfica básica en lugar de las líneas

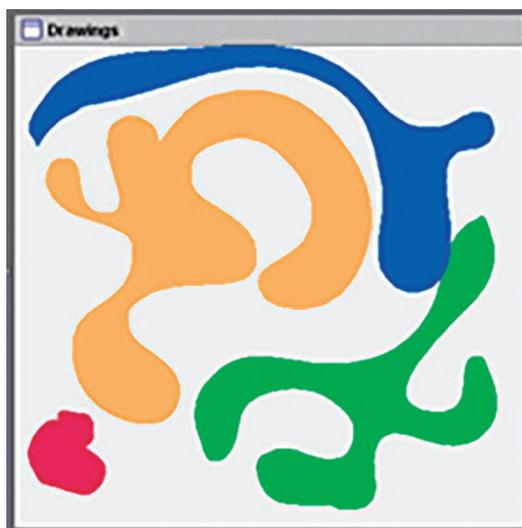


Figura 16.10. | Ventana del director en la versión 2 de IMP1, canvas de control.

de la versión anterior. El pintor-conductor debe crear una forma dibujando su contorno y mientras se está creando la forma, se muestra una línea negra. Tan pronto como el dibujo de la línea se termina de trazar, los extremos de la línea se conectan por una línea recta y la forma consigue el color actual. Según el tiempo que tomó dibujar el contorno, la forma gradualmente aparece y desaparece en la pantalla. En esta representación, aunque la técnica de animación es simple, ofrece transiciones suaves y crea un discurso básico continuo.

La ventana para la colección de ensambles

La interface del conductor tiene también una pequeña ventana (figura 16.11) que se subdivide en dos secciones: la primera es un área blanca rectangular donde todos los músicos en línea están representados como pequeños rectángulos y cada vez que un nuevo músico se registra en el sistema se crea un pequeño rectángulo con un número que aumenta secuencialmente en el centro del área. Los rectángulos se pueden desplazar libremente alrededor del área con la intención de que el conductor pueda organizar a los músicos geográficamente. La ubicación en la pantalla puede, por ejemplo, indicar la ubicación física de los músicos en el escenario o, por otro lado, puede significar importancia o el orden de aparición. Esta decisión se deja abierta para que (intencionalmente) cubra los requisitos del ejecutante. Tan

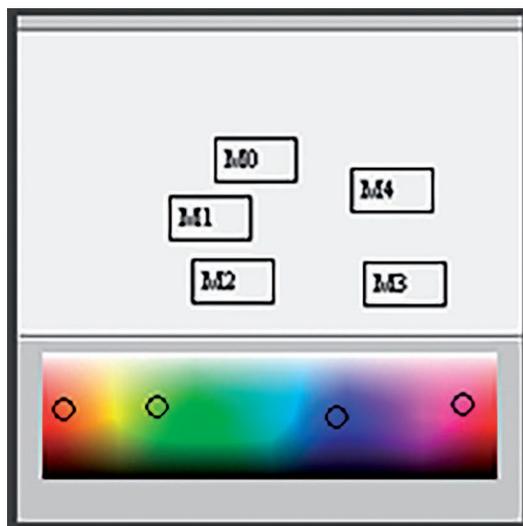


Figura 16.11. | Configuración de ensamble para control de IMPI.

pronto como un músico se desconecta o desconecta el sistema, el rectángulo que lo representa se borra en automático.

La sección inferior es un área rectangular que muestra toda la gama de colores. Los valores del color están organizados de tal manera que se crea un continuo de color similar al arco iris en el eje x. En el eje y se presentan variaciones con valores mayores o menores de blanco y negro, creando un continuo gradual.

La sección inferior se utiliza para crear combinaciones de músicos o subconjuntos, cada subconjunto está representado por un pequeño círculo sobre los colores. Por ejemplo, para crear un subconjunto el usuario hace clic en la ventana superior por encima de todos los músicos que formarán parte del conjunto y, una vez seleccionados todos ellos, el usuario hará clic en la sección inferior del color que se desea representar como subconjunto particular. En ese momento se crea un nuevo círculo sobre el color que se eligió y cada vez que el cursor está sobre un círculo, todos los músicos que componen ese conjunto se representarán con el color particular de ese conjunto en particular.

Como en el caso de los músicos, los subconjuntos se pueden mover libremente alrededor del área; sin embargo, para este caso, hay un efecto porque el movimiento modifica el

color del conjunto y establece el color actual para la pintura. Este punto se describirá en la sección de correlación e interacción.

La ventana para la colección de objetos

La interface del pintor también tiene una pequeña ventana (figura 16.12) donde todos los objetos se recaban y se organizan. Esta ventana se encuentra debajo de la colección de conjuntos y se representa como un rectángulo blanco. Cada vez que se concluye una forma en la ventana del lienzo se crea una versión escalada en la ventana de colección de objetos. Por defecto cada nueva forma se agrega en el centro por debajo de las anteriores; sin embargo, una vez creada, se puede mover libremente alrededor de la ventana para adaptarse a los requisitos particulares del reproductor. Se permite, por ejemplo, estar organizados por la forma, el color o el orden en que se repitan y requieran los materiales.

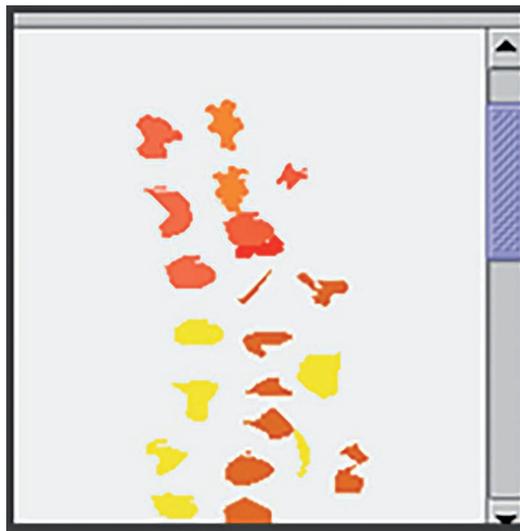


Figura 16.12. | Configuración de objetos para control de IMPI.

Si un pintor-conductor selecciona el objeto, se crea una nueva ventana (figura 16.13). En la ventana se presenta una copia del objeto. Y alrededor de todo el contorno se muestran varios pequeños círculos. Estos círculos representan los puntos finales de control de



Figura 16.13. | Configuración de edición de objetos de IMPI.

las curvas de Bezier que componen la forma; este punto se explicará más adelante en esta misma sección.

Todos los puntos se pueden arrastrar y mover alrededor de la ventana provocando una transformación de la forma original y, una vez que el usuario esté satisfecho, con la variación la ventana se puede cerrar y el nuevo objeto se puede agregar automáticamente a la colección de objetos. Originalmente, esta transformación sería el método utilizado para crear variaciones gráficas y visuales, sin embargo, no fue implementada.

La ventana de control

En la parte inferior de la interface del pintor hay otra sección (figura 16.14) que corresponde al panel de control que se compone de varios elementos diferentes. A la izquierda hay un cuadrado que muestra constantemente el color actual que se usará para pintar, y que evidencia el conjunto que ha sido seleccionado. A la izquierda del cuadrado se encuentra el botón de reproducción que se utiliza para enviar a los músicos la información que se ha recogido desde la anterior pulsación del botón y, a la izquierda hay un segundo botón que no fue asignado.



Figura 16.14. | Panel de control, segunda versión de IMPI.

Finalmente, a la derecha de la representación, un objeto establece la transparencia de la forma y la intensidad de la frase musical. Este objeto se compone de dos elementos vinculados: en la parte superior tiene un área rectangular con una gradiente que va de blanco a negro y por encima de la gradiente se despliega un triángulo rojo que indica la transparencia actual. Por otro lado, en la parte inferior, hay dos rectángulos móviles: un rectángulo indica la dinámica más suave *pppp* y el otro la alta dinámica *ffff*. Con el ratón o con la tableta es posible modificar la ubicación del triángulo y los rectángulos y modificar la dinámica. ¿Cómo? Al mover el triángulo se selecciona una nueva transparencia para dibujar las formas; moviendo los rectángulos se seleccionará un mapa de dinámica diferente. Hay una correlación lineal entre la transparencia del dibujo y la intensidad de la frase musical por defecto, si el objeto es casi transparente la frase musical es lo más suave posible y mientras el dibujo se vuelve más oscuro la intensidad aumenta.

Los ejecutantes

La interface del músico (figura 16.15) está formada por cinco ventanas donde cada una tiene diferentes características; pero, sin embargo, todas las ventanas implementadas muestran la información musical con un tipo particular de representación gráfica. Una vez que el músico inicia el programa no hay nada que establecer porque el cambio y la conexión con el servidor se realizan automáticamente. Esta versión no permite ningún tipo de entra-



Figura 16.15. | Pantalla para el ejecutante, segunda versión de IMPI.

da por parte de los músicos con lo cual el ordenador sólo utiliza el recurso como una pantalla-partitura; sin embargo, hice especial hincapié en la búsqueda de representaciones alternativas de los motivos musicales.

Diferentes maneras de representación musical; una manera natural para crear interactividad

Mientras trabajaba en la segunda versión del sistema IMPI, empecé a elucubrar que si los músicos no tuvieran una *cadena* digital para alimentar la información al sistema deberían, por lo menos, contar con opciones para modificar la información que recibieron del pintor-conductor; sin embargo, la notación tradicional que se utilizó en la primera versión no parecía funcionar como el método apropiado para ello porque las partituras con notación musical tradicional no ofrecen mucho espacio para la variación. Dado a esta situación consideré diferentes posibilidades para resolver la situación. Una de las soluciones fue crear diferentes tipos de representaciones musicales donde cada una le diera al músico un nivel diferente de independencia, que fuera desde el determinismo de la notación musical estándar hasta el hecho de permitir que los músicos crearan sus propias interpretaciones de formas gráficas como partituras abiertas, que considerara la posibilidad de crear partituras para músicos que no saben leer música en partitura. Esta opción funcionaba como alternativa para invitar a todo tipo de músicos a interactuar con el sistema.

Debido a que los músicos tienen la opción de alterar o modificar la información que reciben, o incluso, generar una línea de pentagrama, se produce una interacción natural con el sistema en esta versión porque los músicos no están necesariamente limitados a lo que se presenta en la partitura y pueden interactuar con las mismas condiciones que ofrece la improvisación libre tradicional. Los músicos pueden elegir entre las diferentes representaciones musicales según sus deseos y habilidades y cambiar voluntariamente entre las ventanas. Las representaciones musicales consideradas para esta versión fueron:

La ventana de puntuación La ventana superior de la interface muestra el material que envía el pintor en su representación como notación musical estándar. Esta ventana correspondería a la ventana de la primera versión de IMPI; sin embargo, no se imple-

mentó por completo cuando decidí trabajar en la próxima versión. Es importante señalar que para esta versión los músicos deben saber leer las partituras y no pueden alterar la información recibida.

La ventana de boceto Debajo de la ventana de puntuación hay otra representación del material musical. Esta ventana fue diseñada para preservar la intención musical y la estructura de la improvisación, pero los músicos no están obligados a saber leer música y tienen un marco mucho más grande para desarrollar ideas personales. Esta representación de la música es, en muchos aspectos, comparable a los rollos de pianola; sin embargo, también mantiene similitudes con la ventana de desplazamiento. Las notas musicales se representan con rectángulos donde la longitud del rectángulo representa la longitud en el tiempo y su oscuridad representa la intensidad de la nota. Los rectángulos casi blancos representan notas suaves y los rectángulos negros representan notas más altas. Las notas se desplazan constantemente de derecha a izquierda, como ocurre en la ventana de puntuación y donde la posición del rectángulo en el eje Y representa el tono de la nota. La parte inferior de la ventana indica el registro más bajo del instrumento y la sección alta de la ventana indica el registro más alto del instrumento. La información corre de derecha a izquierda, emulando la lectura de la notación estándar.

La ventana de dibujo Debajo de la ventana de esbozo, entre otras dos ventanas cuadradas, hay un área cuadrada blanca donde los músicos reciben sólo la información gráfica que se pretendía dirigir a ellos. Los músicos no reciben la animación final, sino únicamente los objetos que fueron enviados a los subconjuntos a los que pertenecen. Los músicos deben utilizar esta información gráfica para crear una representación acústica subjetiva de las imágenes. De todas las representaciones esta es la que ofrece una casi completa independencia a los músicos, pero incluso con toda la libertad que ofrece esta opción se espera que al menos una ligera intención de la estructura definida por el conductor se mantenga durante la ejecución.

La ventana Mensajes En esta ventana el músico puede recibir información que no necesariamente tiene una notación simbólica. Podría tratarse de texto escrito. Esta ventana no se implementó cuando decidí dejar de trabajar en esta versión.

La ventana de animación A la derecha de la ventana de dibujo hay otra área cuadrada blanca donde se presenta la versión final de los gráficos. Su función es ayudar al músico a ver el desarrollo de la sección visual en caso de que no tengan fácil acceso a la proyección. Esta ventana no se implementó completamente cuando decidí dejar de trabajar en esta versión.

Mapeo audiovisual

En esta segunda versión las relaciones entre el material visual y el audio comenzaron a ser estudiadas con mucha más seriedad. El hecho de retomar algunos elementos de la primera versión, descartar otros y tratar de crear combinaciones audiovisuales complejas pero coherentes resultó en un siguiente mapeo. Como en la primera versión toda la información fue creada por el pintor, una de las premisas en la construcción de esta versión concibió la idea de que la acción de crear un dibujo implica contenido expresivo que sí puede traducirse al dominio acústico y puse especial interés en el análisis de la velocidad, la presión y el tipo de forma.

Velocidad La velocidad y los cambios en la misma al crear un trazo tienen mucho significado físico y un alto contenido gestual. En la música, el ritmo es probablemente el elemento que se encuentra más directamente asociado con el movimiento del cuerpo y, debido a esta similitud, esta versión de *IMPI* utiliza la velocidad y sus cambios (aceleración) de cuando se crea el dibujo para construir el ritmo de la frase musical. Si el trazo es lento se crean notas largas; si el trazo es rápido se crean muchas notas cortas; por otro lado, si la velocidad es constante se crea un pulso regular, y si hay cambios en la velocidad se crean pulsos irregulares. La calibración de los valores para este parámetro fue producto de una larga exploración.

Color Como se explicó en la sección anterior, el color se asigna a los subgrupos que comparten una idea musical. Cuando el pintor crea un subgrupo se le asigna un color y cada vez que el conductor elige pintar con ese color se está enviando información al subgrupo particular que posee ese color.

Transparencia Como se ha explicado en la sección anterior, la transparencia de la forma puede modificarse mediante un deslizador. La transparencia de la forma determina la intensidad de la frase musical y puede modificar dinámicamente con el valor de presión de la pluma.

Presión Fue al final del desarrollo de esta versión que se implementó la tableta digital. Antes de su implementación, la transparencia visual y, por lo tanto, la intensidad de la frase musical se obtenía asignando el valor en el deslizador; sin embargo, una vez que el sistema tuvo la opción de usar la presión como un parámetro de entrada, se decidió usarlo para esta intención. Por esta razón hay dos maneras de modificar la intensidad de transparencia: directamente en el control deslizante, o cambiando la presión del lápiz sobre la tableta digital.

Trayectoria La trayectoria del trazo se utiliza para derivar los tonos que constituyen la frase musical en conjunción con el ritmo de la velocidad y la dinámica de la presión. La intención de esta versión estaba guiada para evitar las asignaciones que podrían ser demasiado obvias, demasiado complejas para la percepción del escucha o para aquellas que no ofrecían el mismo resultado con la misma entrada. Se desarrolló un algoritmo simple pero potencialmente interesante, que se explicará más adelante.

Duración Inicialmente, el tiempo que se tarda en crear la forma se utilizó como referente para la duración total de la frase musical; sin embargo, después de algunas pruebas estaba claro que la proporción debía ser modificada porque era extremadamente difícil para el pintor crear suficientes trazos para mantener material musical necesario funcionando en el sistema. La proporción se multiplicó entonces por 10, con lo cual, el tiempo que se tarda en crear el trazo genera una frase musical 10 veces más larga en el tiempo.

Tecnología de la segunda versión

El cálculo de altura Para la creación de los tonos en cada frase musical recurrí al análisis del contorno del dibujo. El contorno de cada trazo puede ser visto como una colección de puntos secuenciales (figura 16.16) porque cada vez que se crea una nota nueva se calcula el centro de gravedad actual de la forma. El centro de gravedad es el

punto en el que sería necesario equilibrar una superficie sostenida en un solo punto. Algunas formas simétricas como círculos, elipses y cuadrados tienen su centro de gravedad exactamente en el centro, pero las formas irregulares pueden llegar a tener este punto dentro o fuera de ellas. Obtener el promedio de los valores x y y de todos los puntos que crea la forma del contorno da el centro de gravedad de la representación. Es importante notar que esta técnica no es matemáticamente correcta, pero es absolutamente funcional para nuestra aplicación práctica.

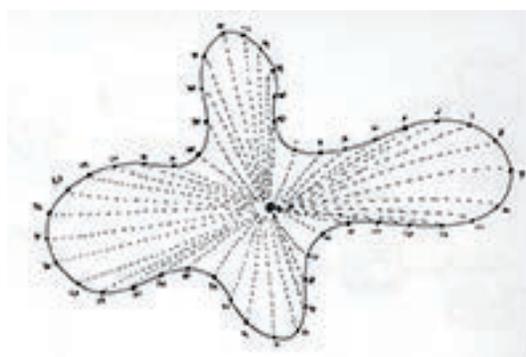


Figura 16.16. | Técnica para el análisis de trazos para la segunda versión.

Cada vez que se creó una nota nueva, se midió una distancia con el centro de gravedad actual y su ubicación. El valor de esta distancia se utilizó para calcular un tono. Por ejemplo, si la ubicación actual estaba cerca del centro de gravedad el tono de la nota era c medio y cuanto más lejos se encontraba el punto del centro de gravedad, más lejos era la nota cromáticamente en relación al c medio generador. Incluso si en la práctica este algoritmo no parecía ser lo suficientemente eficaz, la idea de crear los tonos basados en la forma de las figuras fue un método muy interesante para crear un enlace no tan directo como otros métodos (valores de puntos).

Suavizado Cuando se dibuja una forma con el ratón o con la tableta digital, un reloj interno actualiza constantemente la ubicación actual del dispositivo y todos los puntos están almacenados en una matriz y conectados con líneas rectas. Para muchas aplica-

ciones, la resolución temporal es suficiente para crear la cantidad de puntos necesaria para producir la ilusión de un continuo. Sin embargo, si el dibujo se crea a altas velocidades, la velocidad del reloj no es lo suficientemente rápida y las rectas son observables, dado que la velocidad del reloj es una limitación del sistema operativo, no hay manera fácil de modificarla.

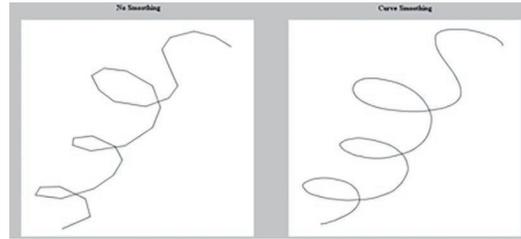


Figura 16.17. | Suavizado de trazos para la segunda versión.

Basándome en el trabajo de Robert Sentinella Robert Sentinella (<http://merganser.math.gvsu.edu/david/reed03/projects/sentinella/>) (figura 16.17), quien desarrolló un Java más suave, descubrí que existen algunas técnicas para suavizar los trazos, este recurso técnico lo implementé en esta versión como método para corregir dibujos rápidos. Este método analiza la distancia entre los puntos y en vez de conectarlos con líneas rectas calcula las curvas de Bezier donde los puntos de control de cada curva se ajustan según la distancia de los puntos. Debido a que la matemática involucrada no es computacionalmente costosa, el método no consume CPU y puede utilizarse en tiempo real. Cuando pude definir la idea de que la transformación de objetos gráficos generaría transformaciones musicales, entonces, surgió el problema de cómo analizar el dibujo dado que la cantidad de puntos que componen una forma gráfica podría ser del orden de miles. Entonces, decidí que la forma necesitaba ser vectorizada para transformar la colección de puntos en algo más manejable.



Figura 16.18. | Método de Newton Raphson para la conversión de trazos en curvas de Bezier.

Utilizando el método de *Newton Raphson* con un programa de Delph de Nils Haeck (<http://www.simdesign.nl/bezier.html>) que aproximándose a una serie de puntos los convierte en una colección de curvas de Bezier y que, a su vez, permite la transformación de imágenes (figura 16.18) el código fue traducido a Java. Para realizar esta traducción conté con el apoyo de Carlos Rocha y una vez que se tradujo me di a la tarea de implementar el método que permite establecer el nivel deseado de aproximación y la suavidad de las curvas con parámetros más manejables.

Técnicas de limpieza La señal generada por el dispositivo de entrada y algunos de los datos utilizados durante el análisis de la información necesitan ser limpiados y suavizados para tener un valor significativo de representación. La información que refiere a la velocidad y a la presión del lápiz sobre la tableta se trasladó a un filtro de paso bajo de primer orden descrito en la siguiente ecuación, en donde t es el tiempo presente y c un coeficiente de filtraje entre 0 y 1.

$$y_t = c * x_t + (1 - c) * y_{t-1} \quad (16.1)$$

Si no hubiera decidido aplicar este filtro la entrada habría sido muy inestable y, por tanto, no habría existido precisión en las mediciones del uso de la tableta.

Técnicas de escalamiento

Algunos de los valores registrados en la tableta eran demasiado lineales para tener un significado significativo, con lo cual, para que estos valores fueran significativos al sistema se aplicó una escala de curva a la entrada.

$$y = \left(\frac{x - i_1}{i_2 - i_1} \right)^c * (o_2 - o_1) + o_1 \quad (16.2)$$

El factor de la curva se calcula mediante un método de ensayo y error y está abierto para ser sintonizado en el futuro mientras el sistema evoluciona y es probado por más ejecutantes.

La tableta digital

Una tableta digital también se implementó para esta versión y la interface funciona tanto con un ratón como con la tableta. Para que la tableta funcione con Java programé un conjunto completo de métodos y clases. Esta librería se basa en la biblioteca JWinTab de Jun Rekimoto (<https://www.sonycsl.co.jp/person/rekimoto/java/jwintab/>) y fue desarrollada en Sony Computer Science Laboratories, Inc. La biblioteca amplía la funcionalidad de Rekimoto y permite tratar la información de la tableta de la misma manera que Java trata con la información del ratón; es decir, usar eventos de la tableta y de los oyentes.

Red

Esta versión vuelve a implementar el motor de red desarrollado para la versión anterior; sin embargo, ahora añade la posibilidad de enviar información a subgrupos particulares y miembros aislados del conjunto.

Observaciones de la segunda versión

Esta segunda versión tiene varias mejoras comparadas con la primera. La codificación de esta versión fue un proceso laborioso que involucró la adquisición de nuevos conceptos y técnicas; sin embargo, hay varias observaciones que se pueden hacer en relación a su perfeccionamiento.

Con la experiencia del taller de Jorn Zorn estaba claro que el sistema incluía características que, aunque interesantes, se implementaban de tal manera que les tomaba demasiado tiempo acceder a la información e interpretar en tiempo real. IMPI está destinado a ser una interface para ser utilizada en un contexto en tiempo real y no hay tiempo para abrir ventanas y arrastrar elementos pequeños si la intención es tener control sobre el material mientras se interpreta la obra musical.

Otro elemento que debe considerarse es la capacidad del sistema para distribuir y desarrollar el material musical para el conjunto porque, aunque los músicos tienen la opción de crear su propia música, el sistema ofrece la posibilidad de crear música para el *tutti* (todos los miembros del conjunto). Esto significa que con una persona que guíe todo el conjunto o con la entrada de una persona individual debería ser suficiente para crear la música de muchos instrumentos. Si componer música es un proceso que requiere diligencia y tranquilidad, crear música para muchos músicos en tiempo real parece ser un desafío complicado dado que si la improvisación coherente como pintor-conductor es, en la mayoría de los casos, un procedimiento complejo, generar improvisaciones interesantes para muchos músicos es incluso mucho más difícil porque su complejidad es multivectorial.

IMPI no intenta ser un generador de música algorítmica; sin embargo, con el desarrollo de esta versión quedó claro que las versiones futuras requerirían mecanismos más precisos para crear más material que lo que se había logrado en las versiones anteriores.

La parte visual de esta versión resultó mucho más rica e interesante que en las versiones anteriores. El uso de formas en vez de líneas, así como el gradual desvanecimiento y la difuminación de las formas, incluyendo la posibilidad de crear formas con diferente nivel de transparencia y la posibilidad de hacer cola de eventos para lanzarlas simultáneamente fueron un hallazgo en positivo y una gran aportación a la mejoría de las imágenes que potencializaron el resultado gráfico haciéndolo mucho más interesante que en las versiones anteriores.

TERCERA VERSION

La experiencia de haber desarrollado la primera y segunda versiones y la posibilidad de haber probado la segunda con un conjunto de músicos en un taller me ayudó a definir las características de la tercera versión. Esta versión está más cerca de la ideal y, aunque todavía hay muchas características que deben ser implementadas, repensadas o modificadas, el cambio más importante fue el acercamiento del material visual al acústico. Uno de los objetivos importantes en la creación de IMPI fue el desarrollo de una interface con las menos posibles definiciones preconcebidas o preestablecidas. En esta versión IMPI intenta automa-

tizar los menos parámetros posibles; sin embargo, la experiencia del trabajo anterior me demostró lo difícil que resulta crear música para muchos instrumentos que se acompañen de una animación visual en tiempo real, definiendo sólo las estructuras básicas de bajo nivel. Esta versión sacrifica parte de este **control completo** del material y predefine algunas relaciones y procedimientos.

De la experiencia anterior estaba claro que la creación de frases de la duración del trazo en sí mismo no era muy eficiente porque el sistema consumía el material mucho más rápidamente de lo que el pintor tardaba en generar el dibujo, provocando grandes secciones de silencio durante el desarrollo de la pieza musical con lo cual, en esta versión, cada trazo genera una frase con duración en proporción al tiempo que le toma al pintor-conductor, pero multiplicado por un factor que posibilite la extensión en el tiempo sin generar silencios. Para lograrlo tuve que considerar la creación de secciones más largas implicando la auto-generación de material; esto se solucionó gracias a que estos materiales se desarrollan determinados por las **esculturas** creadas con dos ventanas nuevas.

Otra de las modificaciones importantes de esta versión fue el cambio en el procedimiento para generar el enlace entre el audio y los elementos visuales; mientras que en las versiones anteriores se utilizaron elementos del dominio gráfico como el color o la transparencia de la forma para crear la música, en esta versión, el audio y los gráficos compartieron un elemento común para lograr su fusión. Estos cambios dieron lugar a un sistema en el que ninguno de los medios depende del otro, sino que cada medio tiene su propio conjunto de técnicas y procedimientos para su desarrollo específico. Se podría decir que esta relación es resultado de la pérdida de los mapeos comprensibles previos; sin embargo, debido a que esta técnica también ofrece una opción de mapeo de bajo nivel, las modificaciones resultan positivas para el resultado musical final.

Esta versión simplifica mucho de los métodos que la segunda versión mostró por ser demasiado compleja para situaciones en tiempo real. Una adición significativa al sistema fue la creación de una ventana que representa la información, utilizando una representación tridimensional. En esta versión hay menos ventanas, y la mano que no había sido utilizada en las versiones anteriores controla algunas teclas de la computadora.

El resultado audiovisual de esta versión de IMPI está mucho más cerca del ideal del sistema porque en la parte gráfica ya se incluyen los filtros y las texturas y se tiene más cuidado en la forma en que se generan los colores. La música también consigue transformaciones más orgánicas y, en el aspecto técnico, muchas partes del código se modificaron para lograr una optimización del uso de los recursos informáticos. A pesar de que todas las técnicas aprendidas y asimiladas durante el desarrollo de las versiones anteriores se implementaron en esta versión (tales como filtros y smoothers), muchas partes del código se modificaron para lograr una optimización del uso de los recursos informáticos, además de reducir el tamaño de las ventanas con representaciones gráficas para permitir el uso de algoritmos gráficos mucho más costosos.

Además de implementar JSML en el código, demostrando su valor trascendental, también se sustituyó Jscore por el renderizador de puntuación utilizado en las versiones anteriores y esta biblioteca se encuentra integrada a JSML porque ofrece la mayoría de las características que IMPI requiere.

Para concluir esta sección me es importante mencionar que esta tercera versión es un trabajo en progreso y que muchos de los cambios aún están en desarrollo y es un sistema absolutamente perfectible en el futuro.

Configuración del ensamble

La configuración del ensamble para esta versión es esencialmente igual a la de la segunda versión en el sentido de que sólo un pintor está realizando el conjunto general y la interacción entre músicos se obtiene mediante métodos acústicos fuera del sistema. El flujo de información en el sistema es unidireccional y no hay posibilidad de que los músicos envíen información al conductor. El código que maneja la funcionalidad de red de la versión anterior se conserva esencialmente en su totalidad para correr esta versión.

La interface del pintor sufrió varias modificaciones en esta versión, mientras que en las versiones anteriores la ventana donde el pintor creó el dibujo era la misma ventana que tenía el material gráfico final. Honestamente, la hipótesis inicial de tener el dibujo actual y el material gráfico anterior en la misma ventana no fue aplicable para esta versión porque los niveles de disociación que se alcanzan entre la huella del gesto y la representación gráfica

pueden llegar a ser tan altos que para esta versión supuse que era más funcional contar con una ventana para rastrear el material original y otra donde se renderizara el resultado.

En las versiones anteriores el espacio de la tableta digital correspondía a toda la pantalla del conductor y esto implicaba que la tableta se subdividía de acuerdo con las distribuciones de las diferentes ventanas de la interface y la configuración no resultaba ser muy eficiente. Cuando el usuario estaba trabajando con una ventana en particular, el espacio potencial del

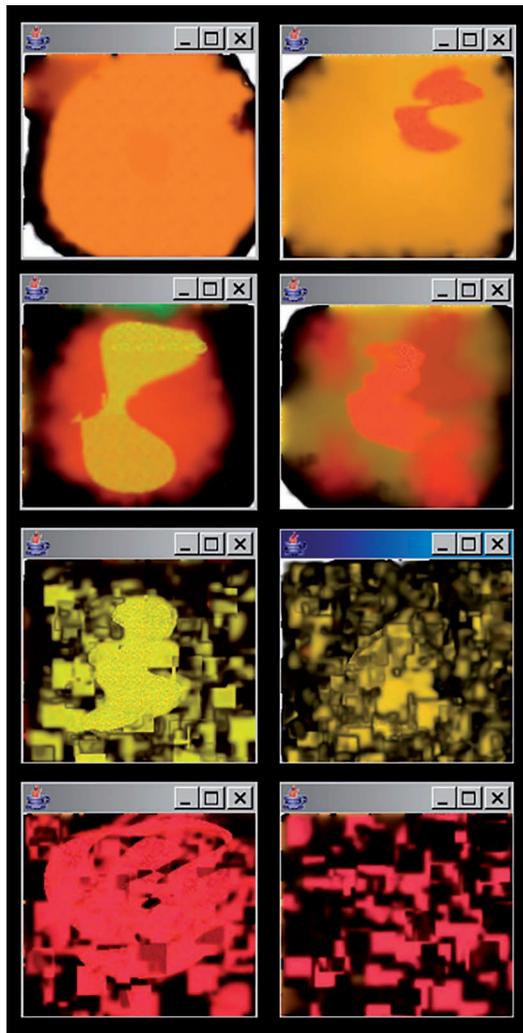


Figura 16.19. | Tomas de pantalla de la ventana de generación gráfica.

resto de las ventanas se desperdiciaba, con lo cual para esta versión consideré utilizar algunas teclas del teclado de la computadora para seleccionar la ventana que se desea y al espacio total de la tableta digital se le asigna esa ventana en particular. De esta manera todo el espacio de la tableta se asigna a la ventana activa permanentemente.

El pintor tiene una ventana con el material gráfico final en la pantalla. Esta ventana (figura 16.19) muestra el material gráfico animado que corresponde al mismo que se debe proyectar durante la interpretación. El pintor-conductor no tiene control sobre él.

La ventana de trazos es una pequeña área blanca (figura 16.20) y cada vez que el pintor inicia un nuevo gesto, la ventana se limpia. Es importante mencionar que cuando se está creando el dibujo la trayectoria que se despliega en la ventana se representa con una delgada línea negra. No es necesario tener un cursor que muestre la posición de la pluma porque el tamaño total de la ventana corresponde al tamaño de la tableta que se selecciona pulsando F1 en el teclado.

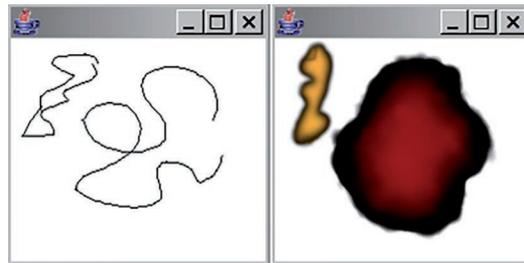


Figura 16.20. | Ventana para trazos de la tercera versión.

La venta de ensambles de esta versión no asigna el color seleccionado actual a un subconjunto particular porque la relación entre el color y la instrumentación no fue eficiente, con lo cual, la técnica para elegir a los músicos que reciben la información también sufrió una transformación en esta versión. Para esta versión de IMPI el usuario tiene que rodear o al menos tocar los rectángulos de los músicos con un golpe y todos los rectángulos se pueden mover a cualquier parte de la ventana creando secciones, de acuerdo con los requisitos del conductor. Cuando se selecciona un músico, se resalta y un subconjunto permanece activo

hasta que el usuario vuelve a activar la ventana que le corresponde. Si no se selecciona ningún subconjunto, la información se envía a todo el conjunto y corresponde a un *tutti* musical.

De manera similar a la primera versión, cada vez que un músico ejecuta el programa en su computadora aparece un rectángulo móvil en la ventana del conjunto, pero tan pronto como el músico cierra el programa el rectángulo desaparece de la ventana. Esta ventana se selecciona con la tecla F2 y no está implementada actualmente.

La venta de regiones El hecho de hacerle clic a los objetos fue una técnica que implementé en las versiones anteriores; sin embargo, el dibujo es la función natural de la tableta digital y hacer clic era ir en contra de la funcionalidad natural de la interface con lo cual, en lugar de hacer clic, esta versión conserva la técnica de dibujar formas para modificar los parámetros que determinan la música y el material visual. Estas ventanas se utilizan para definir las regiones visuales y acústicas. En estas ventanas el usuario puede dibujar formas tridimensionales.

Los detalles sobre la noción de regiones espaciales se explican más adelante. La ventana para definir la región acústica se activa pulsando la tecla F3 y la ventana para definir la región visual se activa pulsando la tecla F4. Actualmente, estas ventanas no están completamente implementadas.

La ventana de color En relación a la ventana de color, para fines de prueba, esta versión conserva la ventana de color que formaba parte de la colección de conjuntos en la versión anterior (vea el panel de color de la ventana de conjunto en la segunda versión); sin embargo, ahora no está directamente asociada con ningún parámetro musical y sólo se utiliza para definir el color principal para la creación del dibujo. El color actual se conserva hasta que se defina uno nuevo, esta ventana se activa pulsando la tecla F5.

Es muy probable que la ventana se elimine cuando el color se convierte en uno de los espacios de transformación controlados con una de las ventanas de las regiones.

Los ejecutantes En relación con los ejecutantes la ventana del pintor-conductor conserva la estrategia de representar la información en varios formatos para permitir al usuario elegir e interpolar entre ellos; sin embargo, estas representaciones han sufrido cambios significativos:

La representación marcada En esta versión, el render de puntuación desarrollado previamente ha sido reemplazado por el paquete Jscore de JMSL [DB04]. La puntuación no se desplaza de derecha a izquierda como lo hizo la versión anterior, sino que ahora la nota que se está reproduciendo en el momento se resalta con un cursor amarillo que se desplaza de izquierda a derecha. Tan pronto como el cursor llega al final de la ventana se crea un nuevo material en ella y se rebobina el desplazador hasta el comienzo del material de la partitura (figura 16.21).

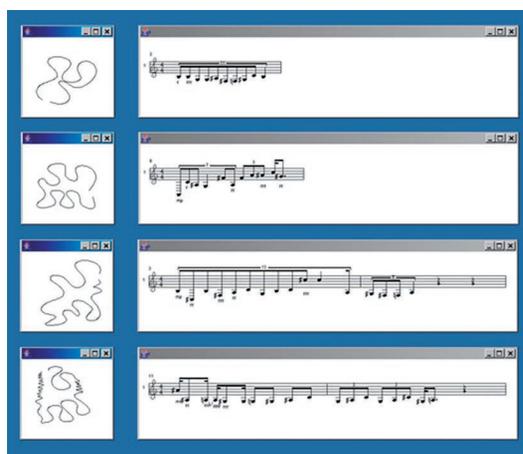


Figura 16.21. | Correlación entre trazos y generación de partitura.

La ventana de gráficos personalizados En esta ventana el músico recibe una representación gráfica de la información que se dirige específicamente al músico, con lo cual múltiples músicos pueden recibir la misma información si pertenecen al mismo subconjunto; sin embargo, debido a que estos subconjuntos están cambiando constantemente, cada músico termina con gráficos diferentes en sus pantallas.

La ventana de material gráfico principal El músico tiene en la pantalla una pequeña ventana donde se presenta el material gráfico principal, por si acaso no tuviera el acceso a la proyección.

Lenguaje audiovisual Hay varias decisiones que aún deben tomarse sobre el lenguaje audiovisual y sobre el mapeo que —finalmente— se utilizará en la versión en desarrollo. Sin embargo, las características importantes ya están definidas y deben ser señaladas. El cambio más importante fue la disociación de las asignaciones directas entre el audio y el material visual. En las versiones anteriores algunos elementos musicales fueron subordinados a parámetros gráficos y ninguno de los elementos musicales afectó de la expresión gráfica. Las relaciones directas fueron descartadas con la creencia de que cada medio debe evolucionar sin interferir con el otro en el mapeo de bajo nivel. El diálogo audiovisual entre las imágenes y la música debe transitar en favor de la creación de tensión y relajación de estructuras más grandes en el tiempo con lo cual el nivel de sincronización reconocible entre los eventos de ambos dominios es el objetivo estético principal y el campo de la investigación.

La tableta digital sigue siendo el método para introducir información en el sistema lo cual significa que la información generada en el momento de crear los trazos continúa como una parte fundamental del algoritmo generador de la información; sin embargo, en esta versión la tableta no se utiliza sólo para generar nuevo material, sino también para transformar la información que ya está presente en el sistema. ¿Cómo se logró esta transición? Esto se obtuvo mediante un nuevo y más complejo paradigma para la creación de material audiovisual que implicó el uso de dos ventanas con diferentes funcionalidades. La hipótesis fue que el control de las relaciones de bajo nivel con la ventana de la lona se extendía por el control de las relaciones de alto nivel con la ventana de las regiones y eso permitía la creación de material coherente e interesante, al mismo tiempo que, tanto el público como los intérpretes o ejecutantes, pudieran entender el lenguaje audiovisual.

La ventana de lona conserva muchas de las características de la versión anterior pero las transformación en los parámetros generados por la tableta digital se mapearon de la siguiente manera:

Velocidad Este parámetro se conserva de la versión anterior. Si la traza es lenta se crean notas largas; Si la traza es rápida se crean muchas notas cortas. Por tanto, si la velocidad es constante se crea un pulso regular; por otro lado, si hay cambios en la velocidad se crean pulsos irregulares.

Duración La duración del trazo define la duración del objeto musical. En el futuro, este objeto se repetirá con una transformación morfológica gradual basada en la estructura tridimensional de la ventana de regiones.

Trayectoria Este parámetro se conserva de la versión anterior. La trayectoria establece los tonos a utilizar. Sin embargo, el valor de la trayectoria se combina ahora con la inclinación para crear el tono final. El código fue mejorado. El centro de gravedad no es el promedio de los puntos anteriores en un momento determinado, sino el punto medio de toda la colección de puntos.

Presión Este parámetro se conserva de la versión anterior. Define la intensidad del gesto.

Inclinación El valor de inclinación se utiliza para definir el registro de la nota que se va a generar. La trayectoria define la nota (C, D, E, F, etc.) pero la inclinación asigna la región (c2, c3, c4, etc.).

Las ventanas de regiones y sus características aún no se han implementado; sin embargo, debido a que fueron concebidas cuando se diseñó esta versión, deben describirse, así que las detallo a continuación.

Una región se refiere a un segmento del espacio total representado en la ventana donde el significado de este espacio resulta distinto para el dominio visual y acústico, pero –para ambos casos– esta ventana representa un espacio tridimensional.

La región se definirá con la tableta digital. Estas ventanas definirán la región que se ve afectada por el trazo creado en la ventana de lienzo, pero debido a que la intención de estas ventanas consiste en establecer el comportamiento de las relaciones de alto nivel que operan durante largos periodos de tiempo, la información efímera, como la velocidad, no se considera en esta ventana existente pero sí se incluyen la ubicación y la presión de la pluma como los únicos dos parámetros que se emplean para crear las regiones donde la ubicación establece los valores x y y. La presión ajustará el valor z del punto específico y el simple hecho de dibujar en la tableta generará un mapa en 3D. Idealmente, el sistema interpolaría puntos para crear formas suaves donde las regiones se visualizaran como esculturas tridimensionales que, a su vez, pudieran ser vistas como macrocontroladores del material e igualmente como el material de representación que se crea con cada nuevo trazo en tiempo real.



Figura 16.22. | Ventana para trazos de la tercera versión.

El uso de texturas y filtros implementados en estas versiones (figura 16.22) demostraba que resultaban métodos más interesantes y expresivos para generar transformaciones orgánicas de la imagen. El hecho de aplicar, repetidamente, el mismo filtro a una imagen en periodos regulares de tiempo ayudó a darle vida al material gráfico, rompiendo con la geometría estática de las versiones anteriores.

Aunque las texturas son fáciles de implementar con la API Java2D, los filtros, por otra parte, requieren más pasos de la programación y se deben diseñar con más cuidado, sobre todo, si se van a utilizar en imágenes animadas, ya que requieren de mucha participación del CPU. Para esto, inicialmente, se empleó la biblioteca de filtros de JH Labs (<http://www.jhlibs.com/>); sin embargo, por razones exploratorias y debido a una ligera incompatibilidad con el nuevo paradigma del procesamiento de imágenes en Java, decidí crear los filtros desde cero.

La observación más importante de la aplicación de esta nueva versión de IMPI consiste en que debemos determinar, después de realizar pruebas, si debemos hacernos un juicio real al respecto considerando, en primer lugar, la idea de utilizar mapas 3D como macrocontroladores después de una exploración y un análisis específico. En el caso de que esta estrategia demuestre ser un método maleable para crear y manipular información, su comportamiento debe extenderse a nuevos dominios: el color en el dominio visual y la instrumentación en el dominio acústico. Los mapas dinámicos donde el significado de la dimensión pueda ser intercambiable también deben ser explorados con este sistema porque si es cierto que las *esculturas* ayudan a visualizar y manipular grandes bloques musicales y visuales, tal vez deban ser tratadas como objetos en el espacio y, entonces, sería interesante poder girar, transformar y desplazar dichos bloques; sin embargo, esta exploración está, por el momento, fuera del alcance de la tercera versión de IMPI.

A pesar de que no podemos incluir el análisis anterior, sí considero relevante que en un futuro próximo encontremos un método que permita incluir la variable de color y las combinaciones de subconjuntos como parte de los espacios dinámicos. En la versión actual estas dos ventanas permanecen sin justificación estética y aunque el hecho de eliminarlas imponiendo los valores que se producen en algunas de las ventanas de la región sería beneficioso, no resulta viable en la práctica ni conceptualmente. En el aspecto técnico es importante implementar algunas características que no son necesarias en el contexto de investigación, pero que son indispensables para su uso en un rendimiento real. En primer lugar, la ventana con el material gráfico debe ser redirigido a un contexto gráfico alternativo que permita enviar una salida de video auxiliar. Este segundo contexto de video debe ser renderizado en una configuración de pantalla completa para proyectarlo sin marcos que perturben la percepción. Además de lo anterior, existen ligeras adiciones que podrían permitirnos que el sistema tenga una pieza mucho más robusta de software. Hay detalles, entre los que se incluye el indicador de ventana activa actual y algunas rutas que se detectan automáticamente (como la resolución y el tamaño de la pantalla), que es necesario considerar si queremos crear una aplicación independiente. Otras observaciones a implementarse en una futura versión serían el uso correcto de la llave para las partituras y otros elementos de notación de partitura estándar para la ventana del pentagrama de los ejecutantes.

CONCLUSIÓN

IMPI es una interface musical de improvisación para un ensamble de músicos acústicos, por lo que el único método real de evaluación del sistema es en un contexto musical dentro de la programación de un concierto o una presentación. Actualmente, sólo se ha probado la segunda versión del sistema, pero tan pronto como se termine la tercera versión, realizaré las pruebas y se hará una nueva evaluación. En el capítulo anterior cada versión se evaluaba separadamente en relación con las otras versiones; sin embargo, en esta sección se evalúan varios aspectos del IMPI tomando como referencia todo el proceso de diseño, implementación y exploración de las pruebas del sistema como una unidad.

Conclusiones del proceso de desarrollo

El sistema IMPI se desarrolló durante el periodo de un año donde los últimos seis meses fueron intensamente dedicados al proyecto. La creación de la primera versión ayudó a mejorar mi desempeño ante la programación de Java y me ayudó en el aprendizaje de algunos de los paquetes y bibliotecas del IDE que se utilizaron para crear IMPI. Durante la creación de la segunda versión tuve una aproximación más directa con el tema visual y aprendí más de técnicas gráficas y de los paquetes de redes. En términos generales se observa un grado significativo de progreso en el diseño del sistema si comparamos los resultados entre la primera y la última de las versiones; por contraste, para la última versión desarrollé técnicas de gráfica avanzadas y estrategias para optimizar el programa y la adquisición de conocimientos técnicos utilizados en IMPI me ayudará para el futuro desarrollo del proyecto.

Conclusiones sobre tecnología

En relación al el ámbito técnico cabe destacar varias observaciones:

- En primer lugar, Java resultó ser eficiente y versátil, el lenguaje y sus bibliotecas funcionaron como se esperaba porque ofrecían la posibilidad de ampliar fácilmente el sistema; sin embargo, como se describirá en la sección “Futuro de IMPI”, en sus futuras implementaciones se considerarán tecnologías diversas que conciban mejoras en relación con la velocidad, la escalabilidad, etcétera.
- En segundo lugar, la tableta digital demostró tener un potencial expresivo, considerando que el dibujo es una actividad común para los seres humanos; sin embargo, también estuvo claro que la tableta no es el método ideal para generar información gestual, especialmente, si hay intenciones de crear información tridimensional.
- En tercer lugar, durante el taller de John Zorn hubo cierta pérdida de información entre las computadoras que demuestran que el motor de la red aún necesita mejoras.

Conclusiones estéticas

La obtención de resultados estéticos interesantes que contenga en sí mismo una propuesta artística consistente es la parte más complicada de un proyecto como IMPI. Luego de plan-

tear varias hipótesis consideré como solución más idónea resolver un sistema como IMPI de manera similar a la forma en que un creador dilucida el proceso artístico de su obra. Por lo que todo el trabajo persiguió el objetivo final de crear una experiencia que pueda expandir la percepción de los receptores del mensaje estético. Confieso que en la primera versión de IMPI no había intenciones estéticas y era un simple prototipo para resolver desafíos técnicos, sin embargo, para la segunda versión ya se intentó crear resultados significativos en relación con el resultado estético y luego de terminar y probar la versión durante el taller de John Zorn pude ver que algunas de las premisas eran correctas y funcionaban bien, mientras que otras no funcionaban como me esperaba y me di a la tarea de poner el ejercicio y la experiencia estética en primer plano, ya que algunas secciones del material creado durante el taller cruzaron la línea entre ser un experimento técnico y ser una obra con significado estético. Desafortunadamente, los breves momentos en que el material musical fue lo suficientemente interesante no alcanzó la consistencia para crear un trabajo orgánico. La tercera versión cuida y ha sido concebida considerando el significado en las ideas transmitidas (contenido), considerando una apuesta conceptual y estética, poniendo atención en la búsqueda de un resultado musical más orgánico. Incluso, ante una versión que no ha sido probada y no está completamente terminada podemos observar que lo que se ha desarrollado hasta ahora en relación al flujo de las ideas, de las posibilidades de creación que ofrece esta versión, se tratará de un material más profundo porque puede crear material audiovisual más expresivo. Es importante enfatizar que IMPI es una interface que tiene sus propias reglas sintácticas y sus propias técnicas de rendimiento, las posibilidades y los límites del sistema tienen que ser aprendidos, dominados y asimilados para desarrollar un lenguaje personal de ejecución y, entonces, evaluar seriamente sus capacidades estéticas. Hasta ahora nadie ha hecho esto y no sucederá hasta que una versión robusta esté disponible de manera profesional. El taller con John Zorn ha sido la única experiencia ahora real en la que el IMPI ha sido probado en un contexto musical concreto y, sin considerar las limitaciones técnicas de la versión e independiente de la ausencia de conocimiento de la interface, estéticamente hablando, dicha experiencia fue un contexto extraordinario para tener un contacto directo con el material audiovisual. La observación más importante de dicha experiencia fue que los gráficos (que aún deben mejorarse) deben coincidir con el nivel de complejidad que un conjunto de ensamble de im-

provisación libre logra generar en el dominio musical, ya que aún existe una inconsistencia entre ambos dominios que afecta la apreciación de la obra en su totalidad estética.

IMPI durante el taller de John Zorn

Durante el taller de John Zorn se realizaron varias improvisaciones usando el sistema IMPI. En el primer ensayo la mayoría de los músicos decidió seguir la ventana de bocetos, esta decisión generó una improvisación de baja densidad porque los músicos produjeron una nota aislada por cada rectángulo que aparecía en su pantalla y no un cúmulo de timbres como yo lo esperaba. La preocupación de los músicos por atrapar el siguiente rectángulo a tiempo los hizo olvidar el resto de los elementos musicales (tales como la intensidad y el tono). En cuanto a mi desempeño, no tuve tiempo suficiente para alimentar la información a la velocidad con la que se consumía el material de pantalla, con lo cual, para mantener una buena velocidad en la generación de información, dejé de cambiar los ensambles y esto significó que todos los músicos obtenían la misma información. Este despliegue de información en la pantalla correspondía a un *tutti* similar al unísono en la ejecución del ensamble que duraba una aproximación en tiempo real de lo que IMPI marcaba y, junto a los músicos en esa situación, yo estaba tan preocupado por generar información para llenar el espacio sonoro que no presté atención a la estructura de la improvisación ni a la sutileza del resultado gráfico. De esta experiencia rescaté que uno tiene que familiarizarse con la plataforma de la misma manera que es recomendable hacerlo con el instrumento antes de ofrecer un concierto.

Después de la primera improvisación tuvimos una discusión y acordamos que los músicos deberían poner más atención en la ventana de dibujo y expandir el rango dinámico porque no eran precisos desarrollando la información sobre la intensidad y, concluimos, que deberían ser más conscientes y estar atentos a la forma general en que la improvisación estaba tomando parte. Después de haber participado de la retroalimentación realizamos otras improvisaciones con resultados que mejoraron muchísimo, dado que los músicos crearon ideas musicales basadas en los dibujos, en lugar de basarse en las notas según el boceto donde la plataforma IMPI operaba más como generador de formas que como productor de contenido musical. Posteriormente, fue interesante ver que durante la presentación el len-

guaje de improvisación se conservaba y, al mismo tiempo, ofrecía la posibilidad de crear sincronía con algunos momentos de silencio, incluyendo secciones individuales.

Al final de las improvisaciones tuvo lugar una discusión que incluyó a John Zorn y a los músicos implicados quienes me dieron sus impresiones y su retroalimentación. Zorn dijo que tomando como referencia su relación con los sistemas audiovisuales para aplicaciones de música en vivo con los que estaba familiarizado, IMPI planteaba uno de los paradigmas musicales más interesantes de representación, ya que atendía la representación múltiple del contenido musical. Los músicos del conjunto describieron en detalle sus impresiones, pero el acuerdo general que más destacaba sobre las opiniones de IMPI fue el hecho de que los gráficos redujeron su atención al rendimiento de otros músicos y a su producción musical e interacción durante la ejecución. Esto puede resultar muy común, ya que la mayoría de los músicos ejecutantes trabajan con partituras que conocen a detalle. La retroalimentación me ayudó a señalar que algunas de las asignaciones eran interesantes y potencialmente expresivas, y que algunas otras partes del diseño de las interfaces de los músicos y del conductor debían modificarse, pero sobre todo me demostró que el sistema requiere un estudio cuidadoso por parte del conductor y de los músicos para que genere resultados interesantes y propositivos, sobre todo, para lograr una interacción consistente entre ellos y una relación más directa con el medio audiovisual. ■ ■ ■



Figura 16.23. Músicos durante las sesiones con John Zorn.

Juum

Este capítulo presenta las propiedades y características de Juum, el software desarrollado por el autor como solución personal para organizar y componer obras de arte sonoras y audiovisuales. El capítulo presenta el contexto de los marcos de composición y herramientas similares a Juum y también las propiedades técnicas de esta pieza de software. Este capítulo también explica los elementos estéticos y conceptuales que llevaron al autor a desarrollar una solución personalizada para su trabajo artístico. Se presta especial atención al concepto de *loop* de retroalimentación arte-ciencia producido cuando se crea software personalizado para obras de arte y estas obras de arte creadas se utilizan para producir nuevas iteraciones del software. Finalmente, el capítulo presenta algunos de los proyectos de arte creados con *Juum* hasta la fecha.

MOTIVACIÓN

Si el medio impregna el significado y las características del medio son dados por su configuración interna, entonces en el contexto digital el código que genera el medio afectará al significado final. Esto es cierto en los diferentes niveles del análisis: técnicamente, definiendo nuestra paleta de posibilidades y restringiendo los recursos dependiendo del software que empleamos para crear una obra; socialmente, mediante la suscripción del usuario a un grupo particular que emplea las mismas herramientas; y pedagógicamente, por la influencia de las obras anteriores de la comunidad en nuestro trabajo.

En un momento histórico donde se desarrolla un código específico para obras de arte específicas podemos sugerir que el código que genera las obras es una parte oculta de las obras mismas. Si el código se altera también se altera la obra de arte, y la obra de arte no existiría si el código —creado con la obra en mente— no se hubiera escrito. Dicho esto es posible ver la acción de la codificación como una extensión de la producción artística. Por tanto, la importancia de desarrollar nuestros entornos de codificación no es sólo técnica, sino también estética y conceptual.

En el último siglo, con la incorporación gradual en la sociedad de las tecnologías digitales, la idea de Wagner del *arte total* se ha hecho por lo menos parcialmente real. Del entretenimiento superficial a las obras de arte sensuales y originales, los creadores están explorando y creando las obras de arte compuestas por integración de medios diversos. El audio, el video, la escultura, la pintura, los gráficos dinámicos, la danza, el teatro, la electrónica, la mecánica, la poesía, la literatura, la ciencia metodológica se encuentran entre una enorme lista de actividades humanas que ahora están interconectadas digitalmente y abiertas para la exploración artística.

Si bien el trabajo en un único medio sigue siendo atractivo y abierto al descubrimiento y a la creatividad, las posibilidades expresivas de interrelacionar diferentes áreas son más atractivas para muchos creadores y ofrecen un espacio más amplio para expresiones originales y desafiantes. Teniendo esto en cuenta y tras un periodo de estudio en el campo de la informática, el autor decidió crear una solución personal que integre los diferentes requisitos que surgieron constantemente en las producciones del autor: generación de sonido, control de dispositivos mecánicos, acceso a bases de datos, control gráfico y, sobre todo, la coordinación temporal entre todas las partes.

Desde el comienzo de la era de la computadora los artistas han estado interesados en emplear aplicaciones informáticas con fines creativos. Un número imponente de código programado se ha escrito en una variedad amplia de lenguajes de programación que abordan preguntas artísticas de una amplia gama de perspectivas. El uso de las tecnologías digitales ha abierto un mundo de espacios de creatividad y ofrece un conjunto de opciones previamente no disponibles (a veces equivocadas). El medio digital ha sido una forma de acelerar procesos y ha ofrecido otros beneficios, limitaciones y paradigmas en la producción de arte.

La intersección del arte y la informática ha producido, entre otras manifestaciones: el código que genera materiales para ser ejecutados por medios analógicos; rutinas que generan motivos pseudoaleatorios o patrones estocásticos, secuencias y motivos; algoritmos que responden de manera trivial o sofisticada a entradas analógicas externas; y código que modela representaciones analógicas o que tratan de imitar comportamientos humanos como la memoria, la expresión y la improvisación, por mencionar sólo algunos.

El enfoque ha variado significativamente a lo largo de los años y hoy en día los artistas digitales coexisten con una variedad de soluciones para la producción artística, incluyendo el uso de lenguajes de programación genéricos como Java, Python, Ruby, C, C++ y LISP; uso de bibliotecas libres y comerciales para lenguajes de programación como jsyn, pyAudio y PortAudio; uso de lenguajes de programación orientados a objetos tales como Processing, openFrameworks, csound, SuperCollider, PureData y MaxMSP; el uso de hardware dedicado, tales como samplers de audio, kits de sensores y equipos multimedia. Con tal cantidad de posibilidades nos enfrentamos no sólo a la cuestión de cómo resolver una solución técnica particular, sino a saber cuál opción nos permitirá centrarnos en la esencia artística. La decisión inevitablemente cambiará el resultado artístico y como se mencionó anteriormente la decisión podría ser vista como parte del proceso artístico.

ENTORNOS CREATIVOS DIGITALES

A lo largo de los años se ha creado un número significativo de herramientas orientadas hacia la creación artística. Después de todo, una gran cantidad de obras de arte comparte la misma tecnología y el mismo paradigma técnico. Dada una serie de iteraciones del proceso, era natural reutilizar las mismas piezas de código, lo que condujo primero a la creación de APIs y bibliotecas y posteriormente a software con una orientación clara y definida hacia la producción artística. Juum se inspira en varias de estas tecnologías e integra algunas de sus propiedades individuales en un solo marco. Como marco contextual es importante mencionar tecnologías similares, inspiraciones y paradigmas de composición relacionados.

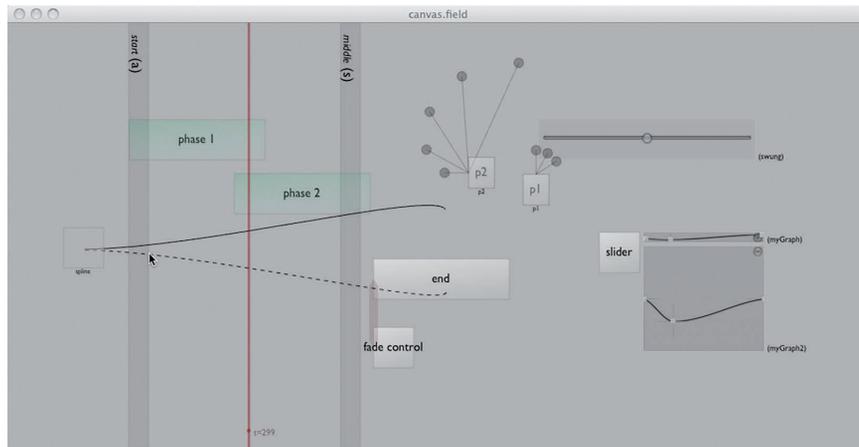


Figura 17.1. | Vista del programa Field.

Field

Field (<http://openendedgroup.com/field/>) es la influencia más directa en Juum; de hecho Juum fue iniciado en un momento en que Field no era de acceso público.¹ Field fue creado por Marc Downie durante varios años mientras era estudiante en el Media Laboratory del MIT. Downie ha producido una variedad de obras visuales utilizando este programa y actualmente se utiliza para todos los proyectos realizados por el OpenEndedGroup compuesto por Marc Downie, Shelley Eshkar y Paul Kaiser (<http://openendedgroup.com>). Field permite la organización del tiempo de fragmentos de código y crea relaciones y conexiones entre cada elemento, así como la manipulación de variables a través de objetos gráficos reduciendo la brecha entre la programación orientada a objetos gráfica y la programación escrita. De esta manera, el código de Field es la partitura y es la pieza. Field permite el control de otros programas como Processing y Max, donde el control de tiempo tiende a ser problemático y menos intuitivo. La comunidad de usuarios de Field sigue siendo pequeña, pero sin duda es una herramienta extraordinaria para crear obras de arte digitales.

¹ Probablemente si Field hubiera sido público en la época que se pensó iniciar Juum, este último no hubiera sido desarrollado.

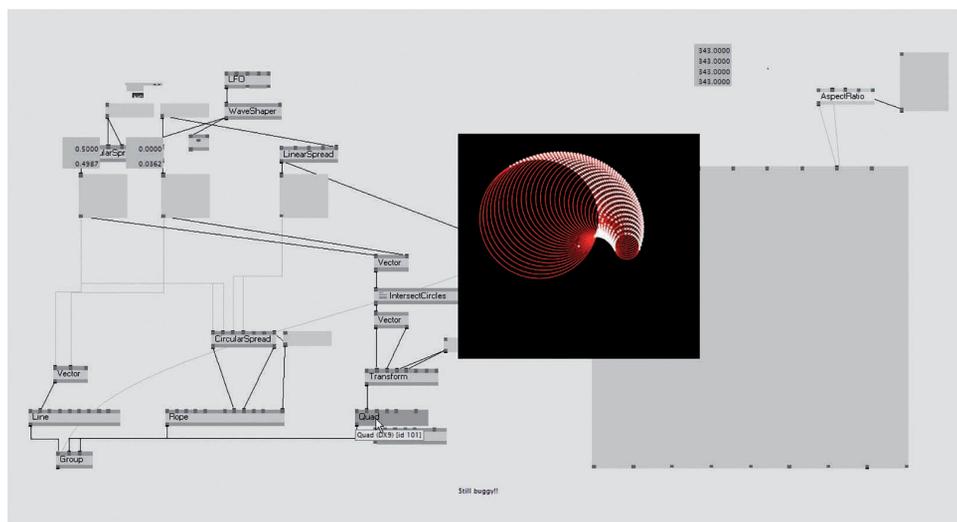


Figura 17.2. Vista del programa vvvv.

Lenguajes gráficos

MaxMsp (<http://cycling74.com/>), PureData (<http://puredata.info/>) y vvv (<http://vvv.org/>) son los lenguajes de programación más representativos de software para el desarrollo de arte interactivo basado en un paradigma gráfico. Más fácil de aprender que la programación en texto, la programación gráfica permite la conexión de objetos gráficos para crear *parches*, que son el código y permiten la modificación de variables a través de objetos GUI. Si bien la creación de pequeñas estructuras y breves arquitecturas en estos lenguajes es sencilla, producir arquitecturas complejas y a largo plazo es, en opinión del autor, un problema y una restricción.²

SuperCollider

SuperCollider (<http://supercollider.sourceforge.net/>) es uno de los lenguajes más populares de programación orientados al sonido. Es un completo lenguaje para la síntesis y manipu-

² Después de varios años de enseñanza y utilización tanto de programación gráfica como de programación de texto, me inclino a pensar que a pesar de que la curva de aprendizaje para la programación de texto es más pronunciada al final la lógica, la claridad y la comprensión son mucho mayores. Por esta razón el autor deja la programación gráfica sólo para cursos introductorios.

lación de sonido.³ Comenzado por James McCartney y sostenido por un grupo de desarrolladores, SC es ahora código abierto con algunas similitudes con csound y otros programas más antiguos creados específicamente para el control de sonido. SuperCollider mantiene la lógica de encadenar pequeños *generadores unitarios* para crear estructuras de generación o transformación de sonido que son controladas en tiempo real o por una partitura numérica en forma de lista de eventos.

La línea de tiempo

El paradigma de la línea de tiempo está presente en una amplia variedad de software. Los editores de video y audio tradicionales se basan en cronogramas estáticos para representar la evolución del tiempo fijo. Este paradigma ha permitido la edición no lineal de materiales. Sin embargo, hay pocos programas con líneas de tiempo no lineales basadas en operadores lógicos que pueden cambiar el orden de las diferentes escenas. Tener tal versatilidad era una de las lógicas fundamentales en la creación de Juum.

DESCRIPCIÓN

Como se mencionó anteriormente, Juum,⁴ es un entorno de composición fuertemente inspirado por el software Field creado por Marc Downie. Juum permite la organización fácil e intuitiva de los comportamientos temporales. Estos comportamientos se conciben como piezas cortas de código en las que se describen eventos de control sónicos, eventos visuales y de control de hardware de forma algorítmica y programática. Cada uno de estos comportamientos (fragmentos de código) se encapsula en una forma geométrica que se puede manipular en una línea de tiempo y área de espacio-campo. Cada uno de estos fragmentos está escrito en un lenguaje de *scripting* condensado, lo que permite una programación ágil y dinámica.

³ Un gran recurso para aprender a programar en SuperCollider es el libro *The SuperCollider Book* (Wilson, Cottle y Collins, 2011).

⁴ Juum significa “ruido” en maya.

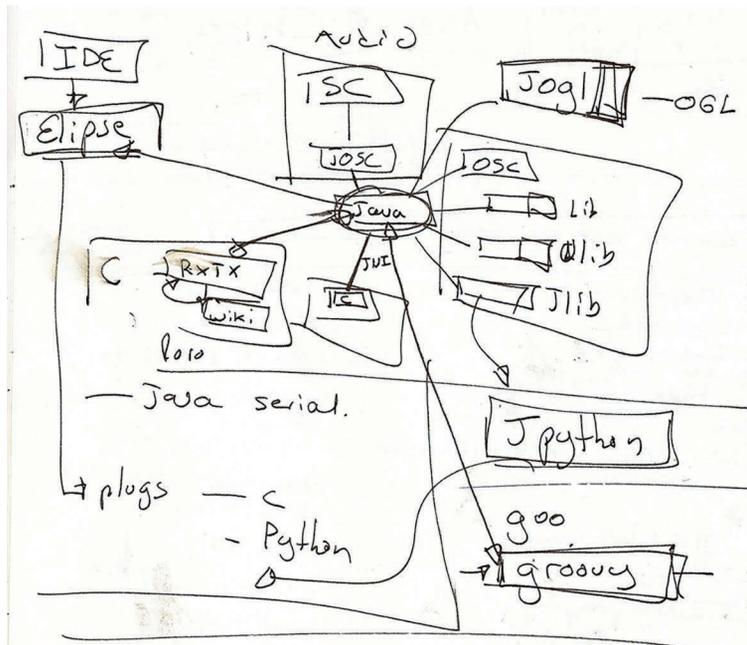


Figura 17.3. | Primer boceto de la configuración del programa Juum.

Estos fragmentos de código pueden ser eventos basados en el tiempo o eventos controlados por la representación espacial en dos dimensiones de la ventana de control. Los eventos de tiempo se representan como formas rectangulares y se interpretan con el tiempo mientras evoluciona la obra. Los eventos espaciales se representan como formas circulares y se interpretan a lo largo de toda la duración de la obra. En ambos casos el tamaño y la ubicación de la forma son variables y personalizables en tiempo real. La forma y la ubicación de cada contenedor gráfico son variables que pueden usarse como parámetros dentro del código encapsulado.

ARQUITECTURA

Juum está escrito en el lenguaje de programación Java y hace extensivo uso de bibliotecas de código abierto. Con el paso del tiempo, conforme nuevas características fueron requeridas, nuevas librerías fueron incorporadas. Las capacidades y opciones específicas de cada biblio-

teca se han incorporado a los métodos preconfigurados que están disponibles en la consola de Groovy. Exponer sólo los métodos de interés mientras que la inicialización de las librerías se hace de manera transparente para el artista simplifica la experiencia de uso.

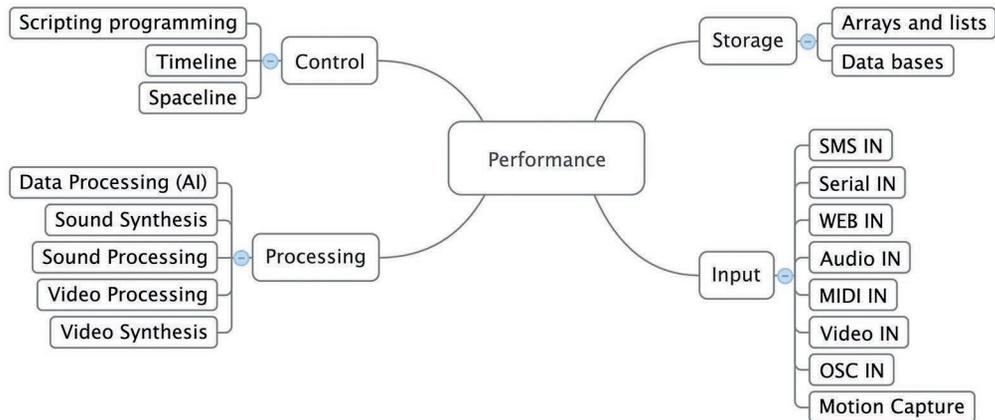


Figura 17.4. | Lógica en la estructura de Juum.

Los materiales, datos y comportamientos se organizan y componen manipulando los elementos gráficos y el correspondiente código Groovy contenido en cada uno de ellos. Lo más atractivo es que cada elemento gráfico puede ser utilizado para describir eventos en medios diferentes, uno para los gráficos y otro para el audio, por ejemplo. Los cambios en dichos objetos se refleja en tiempo real en el *render* de la obra.

Las bibliotecas y el software que Juum utiliza internamente son:

<i>Librería</i>	<i>Funcionalidad</i>
Groovy	analizador de secuencias de comandos
JOGL	motor de dibujo OpenGL
Commons-Math	análisis estadístico
SuperCollider Server	motor de audio
jCollider	punto de conexión SC-Java
Midi Share	control MIDI
NetUtil	control de Open Sound Control

HSQldb	base de datos
JMyron	control de cámara
Opencv	Control de video
RXTXcomm	comunicación serial
SMslib	control de mensajería de teléfono celular
NekoHTML	análisis de la web
Substance	Diseño del GUI

Módulos

Juum contiene varios módulos para recuperar, organizar y acceder a información externa. Dispone de módulos de audio, video, MIDI, OSC, serial y video que permiten la entrada, salida y transformaciones de varios tipos de datos. También tiene métodos para acceder y analizar páginas web, interactuar con mensajes de teléfono SMS y analizar archivos de Motion Capture. Juum tiene una base de datos donde todos los datos pueden ser almacenados y recuperados si es necesario. La base de datos es eficaz cuando se trabaja con grandes cantidades de información que deben ser rápidamente accesibles y catalogables. Los datos recuperados y generados también pueden ser analizados estadísticamente para ayudar al usuario a inferir propiedades y singularidades de la información.

Interfaz gráfica de usuario

Gráficamente, el software está organizado en cinco ventanas. La ventana central más grande es la línea de tiempo y espacio. Aquí es donde los objetos se agregan, se editan y se mueven para organizar la composición. El área funciona simultáneamente como una línea temporal donde una línea vertical roja se desplaza de izquierda a derecha durante el progreso de la pieza y también como un espacio bidimensional donde la ubicación de los elementos crea relaciones espaciales entre los elementos circulares. En su función de línea de tiempo, las acciones de cada caja sólo están activas mientras la línea de tiempo interseca al objeto. Al primer momento de intersecar al objeto gráfico se le llama función de *inicio*; durante su recorrido se ejecuta una función de *loop* y al terminar se ejecuta una función de *final*.

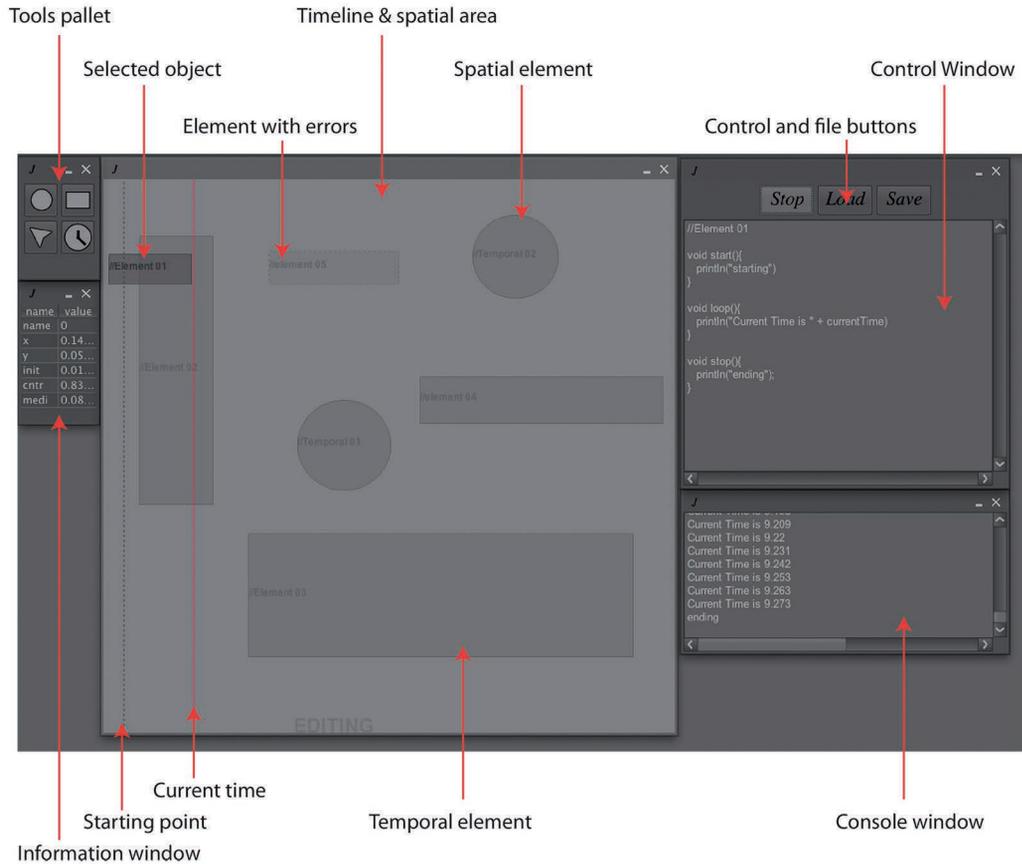


Figura 17.5. | Elementos del entorno gráfico de Juum.

La ventana de control colocada arriba a la derecha muestra el script correspondiente al objeto seleccionado actualmente en la ventana principal. Aquí es donde el usuario escribe el código que tiene que ser evaluado para ese objeto en particular. El código puede modificarse en tiempo real e incorporarse a la composición. Los problemas de compilación con el objeto geométrico son reflejados por cambios en su color y línea de contorno. Esta ventana también tiene los botones de inicio, detención y administración de archivos, así como opciones como pantalla completa y monitor de datos del puerto serial.

Juum comparte varias propiedades con otros lenguajes de programación que históricamente han demostrado ser intuitivos y fáciles de usar. Juum tiene una ventana de consola (abajo a

la derecha) donde el usuario puede leer la salida de texto del programa. Esto es bueno para obtener información de texto del software y también para propósitos de depuración.

Juum tiene también una ventana de paleta de herramientas (arriba a la izquierda) donde el usuario selecciona la herramienta a emplear. La versión actual incluye cuatro herramientas: añadir elementos temporales, añadir elementos espaciales, seleccionar un elemento ya creado y cambiar el momento actual. Una ventana de información (abajo a la izquierda) proporciona detalles sobre la ubicación de tiempo/espacio de cada elemento. Los números se pueden cambiar directamente en esta ventana y los resultados se reflejan en los objetos gráficos de la ventana principal.

El programa se ejecuta desde un comando en terminal, si se desea y por medio de banderas de entrada, Juum también puede generar otras dos ventanas adicionales. La ventana de video muestra el monitor de video y algunas propiedades de la cámara como saturación, tono y contraste. La ventana de lienzo muestra los gráficos dinámicos, que se muestran en pantalla completa si se configura un segundo monitor.

CÓDIGO DENTRO DE JUUM

Los usuarios interactúan con el software Juum de dos maneras: colocando y distribuyendo los objetos geométricos en la línea de tiempo/espacio, y escribiendo código para cada uno de los objetos gráficos individuales. El lenguaje utilizado para escribir el código de los objetos es Groovy (König *et al.*, 2009). Groovy es un lenguaje scripting con características como *closures* y constructores y tipos dinámicos que lo hacen ágil y compacto. Es una opción ideal para desarrollar pequeños fragmentos de código en los que no se necesitan declaraciones de variables o comprobaciones de tipos. El siguiente ejemplo muestra una rutina básica simple con elementos de impresión en los momentos inicial y final y un contador variable que aumenta a una velocidad controlada dentro del método de bucle.

```
1. counter = 0
2. def start(){println "start here"} def loop(){println counter++
   }
3. def stop(){println "bye"}
```

USO DE JUUM EN CONTEXTOS ARTÍSTICOS⁵

Juum ha sido utilizado en varias presentaciones y proyectos y ahora es parte integral de las improvisaciones solistas del autor para piano y electrónica. El Dúo Juum, compuesto por Gabriela Villa en viola y poesía y el autor en la parte electrónica, también emplea esta herramienta regularmente.

La primera iteración del software fue desarrollada como un proyecto para el control de curvas paramétricas durante un curso de Audio Digital en el Centro de Arte Digital y Medios Experimentales de la Universidad de Washington. Este boceto inicial sólo tenía un control estático para eventos de sonido y no tenía ninguna manipulación en tiempo real o control de secuencias de comandos. Se incluyeron la entrada/salida MIDI y el primer generador de gráficos. La segunda versión fue utilizada en el Seattle Latin American Music Festival por el Dúo Juum en septiembre de 2007. En ese momento, el software permitió la generación de formas geométricas básicas en el sonido y la manipulación de tonos de muestras pregrabadas utilizando un teclado MIDI.

En marzo de 2008 se creó una nueva iteración que incluye ahora el analizador de secuencias de comandos Groovy, el análisis estadístico, la entrada en serie y el acceso directo a las definiciones de sintetizador de SuperCollider. El nuevo módulo de sonido permitió un mejor control de los eventos de sonido. Esta versión se utilizó durante la composición, producción y presentación del CD del Dúo Juum VÉRTICE. Con modificaciones menores esta versión fue utilizada en la improvisación de piano y electrónica durante un concierto de DXARTS en la Universidad de Washington. Para ese desempeño el autor usó un dispositivo que detecta el movimiento de la muñeca y este movimiento corporal fue utilizado como otra capa durante el análisis computacional.

En mayo de 2008 se incluyó la entrada de video y un segundo generador de gráficos más desarrollado para el proyecto *Just a Minute*, un trabajo colaborativo para el espectáculo *Dance in the Digital Domain*. Esta colaboración artística de María Cynthia, Devin McDer-

⁵ Se sugiere revisar los otros capítulos del libro, pues gran cantidad de las propuestas artísticas fueron creadas utilizando Juum.



Figura 17.6. | Uso de *Juum* durante un concierto en el Chapel Performance Space, Seattle, EUA.

mott y el autor explora las dualidades reales y metafóricas entre los escenarios virtuales físicos y proyectados gráficamente, así como los sonidos acústicos y electrónicos.

El software fue utilizado sin modificaciones para una improvisación de piano y electrónica en el Festival de Pixilerations en Providence, Rhode Island, en octubre de 2008. El mes siguiente, *Conjuration of the Memory* fue realizado por el Dúo Juum. Para esta obra la configuración escénica, los movimientos de la viola a lo largo del pasillo, y la ubicación de dos pianos es artísticamente importante. Los objetos gráficos espaciales y variables globales se incluyeron como solución técnica para las exigencias artísticas de la obra. Esta fue también la primera vez que las transformaciones de audio en tiempo real se organizaron y distribuyeron a lo largo de largas secciones de la obra. En abril de 2009, Juum fue utilizado para una improvisación de piano y electrónica durante el concierto de DXARTS en el Chapel Performance Space. Para este concierto no hubo modificaciones técnicas, pero el software se probó con una capa y organización de audio mucho más complejas.

PROYECTOS REALIZADOS CON JUUM

Las siguientes imágenes muestran capturas de pantalla de diferentes *partituras digitales* creadas con el software Juum. Estas imágenes dan una idea de la estructura, forma y organización de cada obra de arte. Las piezas presentadas son:

Axial La primera captura de pantalla (figura 17.7) muestra la partitura de la versión final del trabajo *Axial*. La sección superior muestra el control de los solenoides; la región inferior muestra el punto de partida de los archivos de audio. La línea central representa el control de mezcla de audio.

Tokyo's concerts La segunda captura de pantalla (figura 17.8) presenta una ventana de prueba usada para el control de motores de corriente continua durante los conciertos en Tokio. Esta versión demuestra gráficamente la distribución de los diferentes motores a lo largo del espacio. Esta captura de pantalla en particular muestra 12 motores distribuidos en dos plantas.

Conjuro de la Memoria La tercera captura de pantalla (figura 17.9) muestra la estructura del concierto *Conjuro de la Memoria*. Esta pieza para voces pregrabadas, sonidos de piano, sonidos de viola, y piano y viola en vivo, tiene una duración de 45 minutos. La partitura coordina las diferentes capas de sonido al mismo tiempo, lo que permite la espacialización del sonido de la viola. El elemento temporal (objeto circular) se utiliza para representar la ubicación física del ejecutante de la viola en la sala.

Piano solo La cuarta captura de pantalla (figura 17.10) muestra el material utilizado durante la improvisación de piano en el *Chapel Performance Space*. El material presenta un objeto de control de audio largo y varios elementos cortos donde se generan diferentes tipos de procesos de audio y variaciones utilizando la información MIDI registrada con un PianoBar.⁶

Just a Minute La quinta captura de pantalla (figura 17.11) presenta la partitura usada durante la colaboración de *Just a Minute* con María Cynthia Anderson y Devin McDermo-

⁶ El PianoBar es un dispositivo inventado por D. Buchla en 2001 y comercializado por Moog, que lee el movimiento de las teclas de un piano acústico regular y traduce el movimiento en formato MIDI. Desafortunadamente el dispositivo se dejó de producir.

tt, como parte del concierto en 3D. Siete secciones de menos de un minuto comprenden la pieza. Dado que la duración específica de cada sección podría variar en cada ejecución, el punto en el tiempo se puede actualizar manualmente utilizando números en el teclado del ordenador. Para este rendimiento, Juum controló las transformaciones de video de un *feed* de video en tiempo real apuntando al bailarín, al mismo tiempo que generaba sonidos ambisonics electrónicos basados en la entrada MIDI.

Piece for Computer and Viola La última captura de pantalla (figura 17.12) presenta el primer uso de Juum en un concierto. Juum se utilizó para generar un fondo visual básico al mismo tiempo que generaba la síntesis de sonido basada en la proyección plana 2D de una trayectoria 3D. Esta versión aún no tenía la plantilla gráfica final y tenía una paleta de control muy limitada.

CONCLUSIONES

Se han presentado las características, propiedades y singularidades del software Juum, así como su evolución a lo largo del tiempo y los contextos artísticos en los que se ha utilizado. Debido a su naturaleza, Juum será siempre un trabajo en progreso. Sin embargo, también está claro que se beneficia del bucle de retroalimentación de la tecnología de arte que está ocurriendo con el constante uso artístico y actualizaciones de este marco. Desde el punto de vista técnico es necesario mejorar, optimizar y ampliar varios elementos. También hay algunos elementos GUI que no son ideales para situaciones en vivo y que será el cambio en versiones futuras. ■ ■ ■

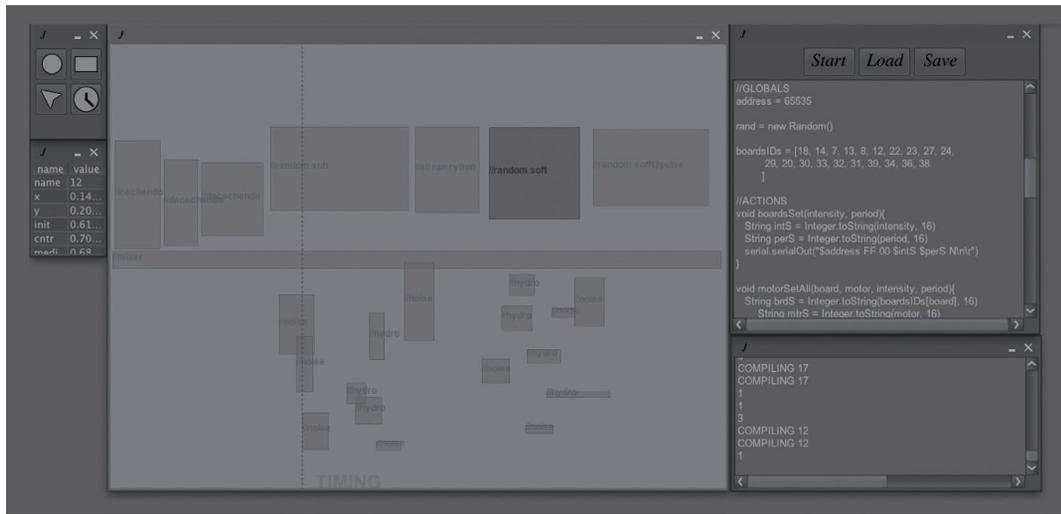


Figura 17.7. | Partitura de Axial creada con Juam.

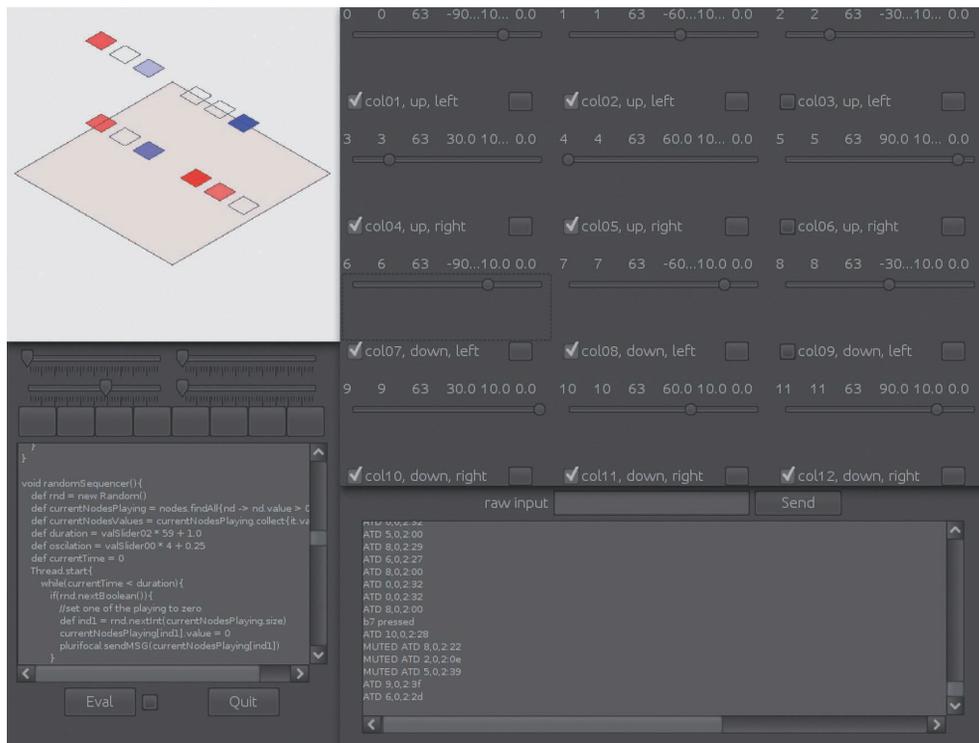


Figura 17.8. | Partitura para los conciertos de Tokyo creada con Juam.

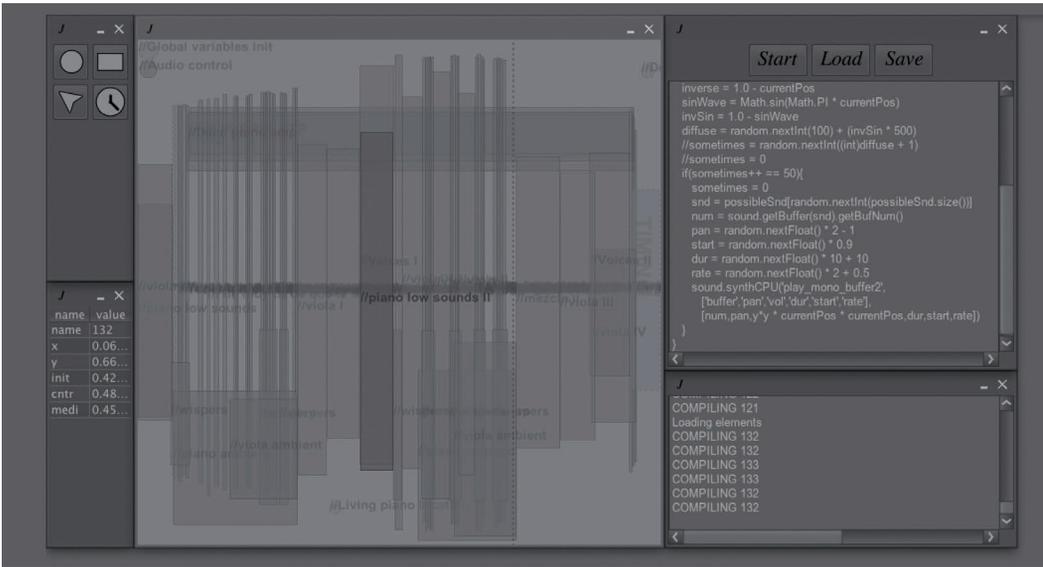


Figura 17.9. | Partitura del concierto *Conjunto de la Memoria* creada con Juam.

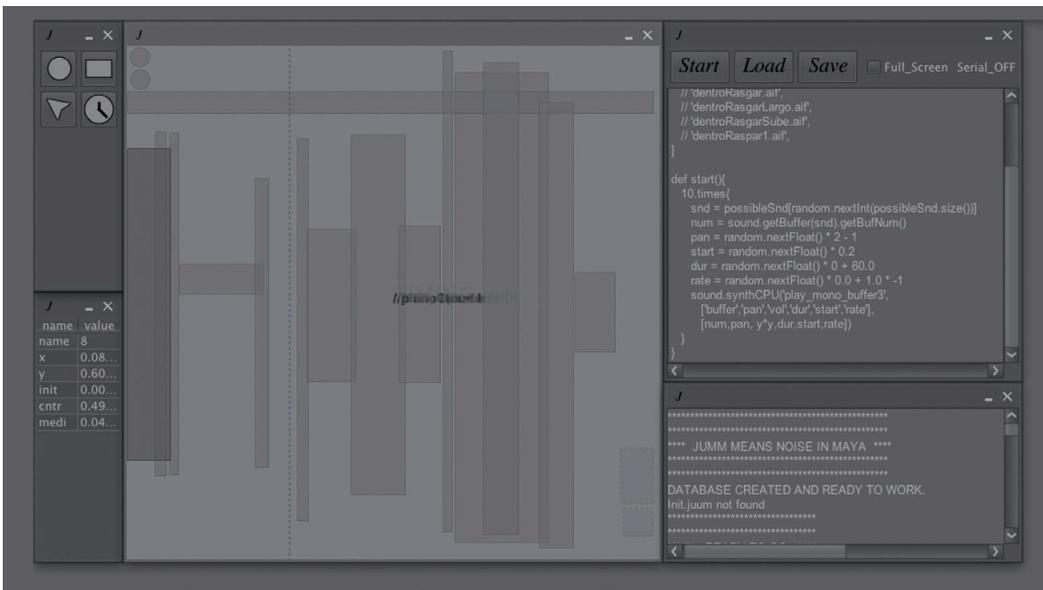


Figura 17.10. | Partitura para las improvisaciones para piano solo creada con Juam.

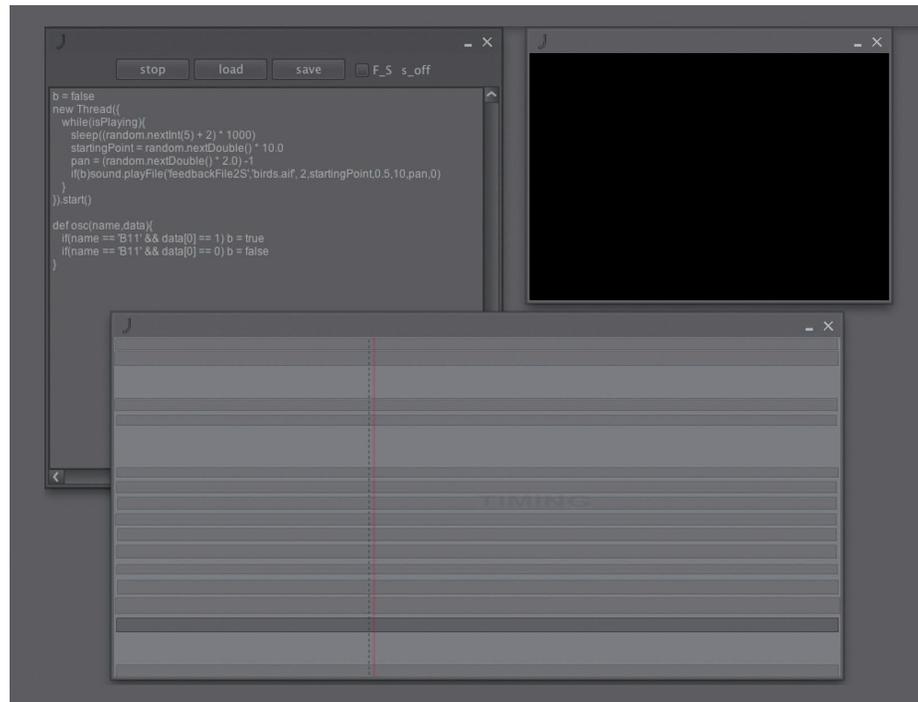


Figura 17.11. | Partitura para el concierto de danza, imagen y sonido creada con Juum.

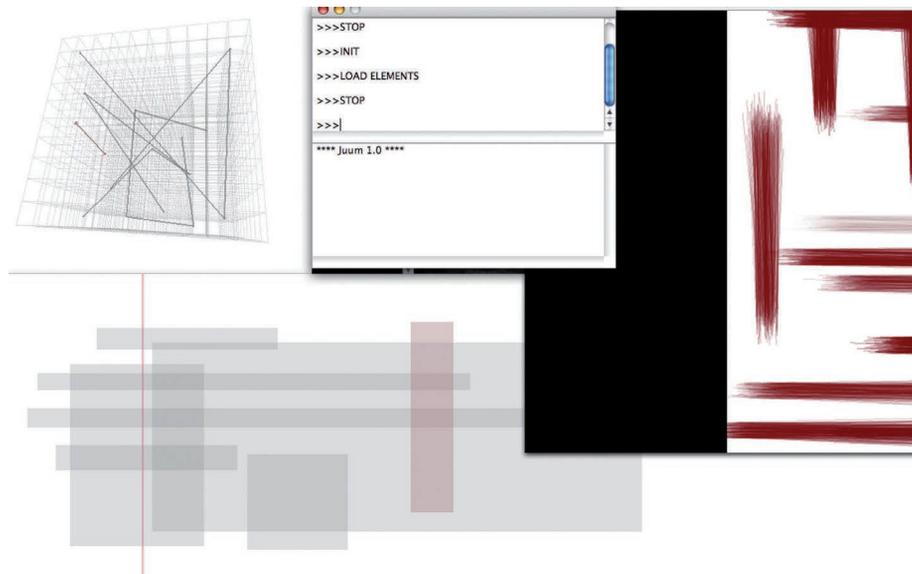


Figura 17.12. | Partitura del concierto para viola y computadora creada con la primera versión de Juum.

Plurifocal sound controller

DESCRIPCIÓN

Con el fin de crear trabajos basados en el concepto de la música plurifocal fue necesario diseñar y fabricar una herramienta que permitiera al autor probar y desplegar trabajos plurifocales. El Plurifocal Events Controller es un dispositivo electrónico personalizado que permite la generación de sonido analógico repartido en grandes espacios físicos. Objetos, recipientes, frascos, paredes, ventanas, etc. pueden ser golpeados, frotados, cepillados o rayados con solenoides y motores DC modulados. Un juego con estos dispositivos electrónicos permite un control preciso del equipo mecánico. El Plurifocal Events Controller fue diseñado, construido y programado por el autor. Cada módulo tiene un microcontrolador principal, un módulo de comunicación inalámbrica, un conjunto de reguladores de voltaje y un conjunto de puertos de entrada y salida. El módulo funciona en conjunto con una tarjeta de alimentación y el conjunto permite el control de motores y solenoides. Con una computadora y utilizando un protocolo personalizado, una colección de estos módulos puede ser controlado de forma inalámbrica para generar eventos sincronizados distribuidos a largas distancias. Cincuenta módulos fueron fabricados durante 2011 y utilizados en varios proyectos, entre ellos *Axial* (descrito en el capítulo 12). Hasta ahora se han creado dos versiones diferentes; la primera funcionó como una prueba de concepto y la segunda se ha utilizado en varias instalaciones que difunden sonidos analógicos en áreas de diferentes dimensiones.

Motivaciones para crear el Plurifocal Events Controller

Como se explicó anteriormente, la principal razón para crear el Plurifocal Event Controller (controlador de eventos plurifocales) fue la necesidad de tener un equipo que permitiera el control preciso de eventos analógicos sonoros extendidos a distancias de hasta 100 metros entre cada nodo. Las opciones comerciales disponibles no resolvían los requisitos o eran extremadamente costosas y aún así había que hacer adaptaciones. Construir una solución personalizada terminó siendo la mejor solución, especialmente porque se introdujo como parte del proceso creativo.

Alrededor de 2004 la tarjeta Arduino se hizo popular en la comunidad artística. Si bien es un concepto de interés, está claro que los desarrolladores sacrificaron el potencial del microcontrolador AVR para hacerlo accesible al público en general. Frustrado por lo difícil que era generar un control de tiempo paralelo con el Arduino (al menos con la versión disponible en aquel momento), el autor decidió obtener una mejor comprensión del uso de los microcontroladores y hacer la programación directamente en el lenguaje C.

TECNOLOGÍA DEL PLURIFOCAL EVENTS CONTROLLER

El Plurifocal Events Controller utiliza un conjunto de circuitos integrados estándar y componentes electrónicos secundarios tales como reguladores de voltaje, condensadores, LED, resistencias y relojes. El núcleo del sistema es un microcontrolador y un módulo ZigBit. Todos los componentes están montados en una placa de circuito impreso de dos lados diseñada en el software Eagle y fabricada externamente en una fábrica de tarjetas PCB que utiliza la serigrafía profesional, el método comercial principal para la producción de gran volumen.

Los elementos centrales de la placa son dos circuitos integrados, un AVR ATmega168 y un AVR ZigBit Amp. El primero funciona como el control principal, manejando la modulación de ancho de pulso (PWM) de todos los puertos de acuerdo con los datos recibidos por el módulo ZigBit. Este último maneja la conexión inalámbrica mediante el establecimiento de

comunicación a la placa maestra en el inicio y la actualización de datos en vivo enviados desde el ordenador.

Con el fin de conectar ambos circuitos integrados y crear una red inalámbrica, ambos dispositivos tenían que ser programados utilizando las bibliotecas ofrecidas por el fabricante. En el transcurso de este proyecto artístico se programaron y probaron varias versiones hasta que surgió una versión robusta y eficiente.

Microcontroladores

Los microcontroladores son circuitos integrados que encapsulan una variedad de unidades que incluyen una CPU, memoria flash, SRAM, comparadores analógicos, convertidores analógico-digitales, convertidores de digital a analógico, generadores de tiempo, EEPROM, USART, unidades de lógica aritmética y funcionamiento general. Registros digitales que están asociados con puertos de entrada y salida. Con un conjunto de instrucciones enriquecido estos dispositivos pueden programarse para ejecutar una variedad de funciones digitales. Debido a su bajo costo, su pequeño tamaño y su potencia y velocidad relativas son una solución ideal para sistemas embebidos donde la lógica secuencial ha sido predefinida y donde el uso de una computadora es excesivo. Sus puertos de entrada y salida los convierten en una solución ideal para crear puentes entre el mundo digital de las computadoras y el mundo analógico familiar.

El término *computación física* describe la esencia de tales dinámicas, en las que los dispositivos digitales incorporados interactúan con objetos reales como luces, motores, sensores, perillas y diversos aparatos y dispositivos físicos. Es natural que el potencial de los microcontroladores atraiga a artistas interesados en experiencias interactivas donde se requiera la retroalimentación del público y la generación digital de material. Los microcontroladores tienen que ser programados primero para establecer las reglas y la lógica secuencial del dispositivo. Hay una inmensa variedad de microcontroladores y lenguajes de programación, pero muchos microcontroladores se basan en el estándar C. Es el caso de los microcontroladores AVR de 8 bits. El ATmega168 utilizado en el Plurifocal Events Controller pertenece a esta categoría. El ATmega168 fue una buena opción para nuestro módulo

porque es un dispositivo de 20 MHz con 16 KB de memoria, tres temporizadores/contadores, seis canales PWM, ADC de 10 bits de 10 canales, USART y 23 líneas de E/s programables¹

ZigBee y BitCloud

Tener una conexión inalámbrica entre dispositivos era importante para nuestra propuesta artística porque queríamos controlar dispositivos distribuidos a largas distancias. Una solución de cable habría sido costosa, difícil de mantener y difícil de instalar y transportar. Por tanto, estudiamos las opciones inalámbricas en el mercado y decidimos utilizar la tecnología ZigBee.

ZigBee es un estándar público creado en 2003 por un consorcio de empresas de la red. Para muchas aplicaciones, los protocolos Wi-Fi y Bluetooth no son buenas soluciones. Wi-Fi requiere grandes infraestructuras. Bluetooth fue diseñado para la comunicación a distancias cortas de menos de unos 20 metros. Ni Wi-Fi ni Bluetooth permiten la creación de redes de malla; ZigBee permite tal configuración, donde los nodos pueden retransmitir mensajes, permitiendo la propagación de mensajes a distancias más largas. Es una opción de bajo consumo de energía, con una tasa de bits de 250 KB por segundo. Está pensado para aplicaciones industriales y de consumo, incluyendo equipos médicos, sistemas de gestión de tráfico, automatización del hogar y otras situaciones que requieren bajas tasas de transferencia de datos a distancias medianas. La capa de red ZigBee admite topologías de estrella, árbol y malla. En una topología en estrella, la red está controlada por un único coordinador. Todos los demás dispositivos se comunican directamente con el coordinador. En las topologías de malla y árbol, la red puede extenderse con el uso de routers. Los ruteadores transfieren datos y controlan los mensajes a través de la red, permitiendo la comunicación peer-to-peer completa².

BitCloud es el nombre comercial de la biblioteca API construida sobre el protocolo ZigBee de Atmel Corporation. BitCloud puede utilizarse en todos los dispositivos ZigBit de esta empresa, incluyendo ZigBit (ATZB-DK-24), ZigBit Amp (ATZB-DK-A24) y ZigBit 900 (ATZB-DK-900). BitCloud es un conjunto de funciones C que permite a los usuarios crear aplicacio-

¹ ATMEL (2009a), Mega 168 Datasheet. ATMEL.

² ATMEL (2009b), *SerialNet User Guide*. ATMEL

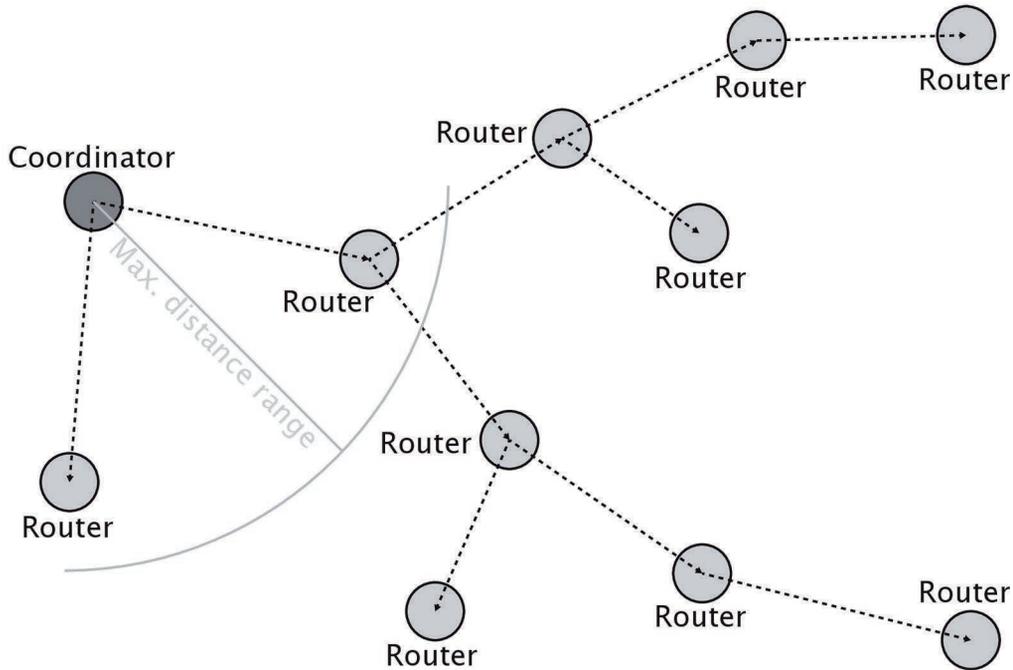


Figura 18.1. Topología de malla de una red BitCloud como es utilizada en el PEC.

nes ZigBit enteras sin empezar desde cero. Además, tiene una capa superior llamada SerialNet que permite a los usuarios llamar a varias de las funciones *c* con un protocolo de serie ligero. SerialNet fue utilizado en las primeras versiones de desarrollo de nuestro módulo. BitCloud en *c* se utilizó en las últimas versiones para tener un mejor control de la comunicación y de las redes de malla³.

FABRICACIÓN DE LA ELECTRÓNICA

La creación del Plurifocal Events Controller fue un proyecto iterativo de varios meses. Varias versiones fueron creadas a lo largo de varios meses. Como se mencionó anteriormente, el circuito fue diseñado con el software Eagle, que permite la generación de esquemas electrónicos. De los esquemas se creó un diagrama de pistas. Éste se puede imprimir

³ Idem.

para crear tablas de hobby utilizando técnicas de grabado químico o se pueden guardar en el formato Gerber. Este formato se utiliza para fresadoras CNC o para fabricación industrial.

En nuestro caso, produjimos un prototipo con una máquina CNC y posteriormente fabricamos las tarjetas con serigrafía profesional. Para todas las versiones producidas los componentes electrónicos se montaron a mano y se soldaron uno a uno con la ayuda de lupas.

Prototipo

Con el fin de probar nuestra tecnología, el primer paso fue la creación de un prototipo de prueba. El diagrama de dos capas se creó con Eagle y luego se imprimió con el Circuit Board Plotter en la universidad. Se empleó una cantidad significativa de tiempo uniendo a mano ambos lados de los orificios de conexión entre capas. La mayoría de los componentes eran

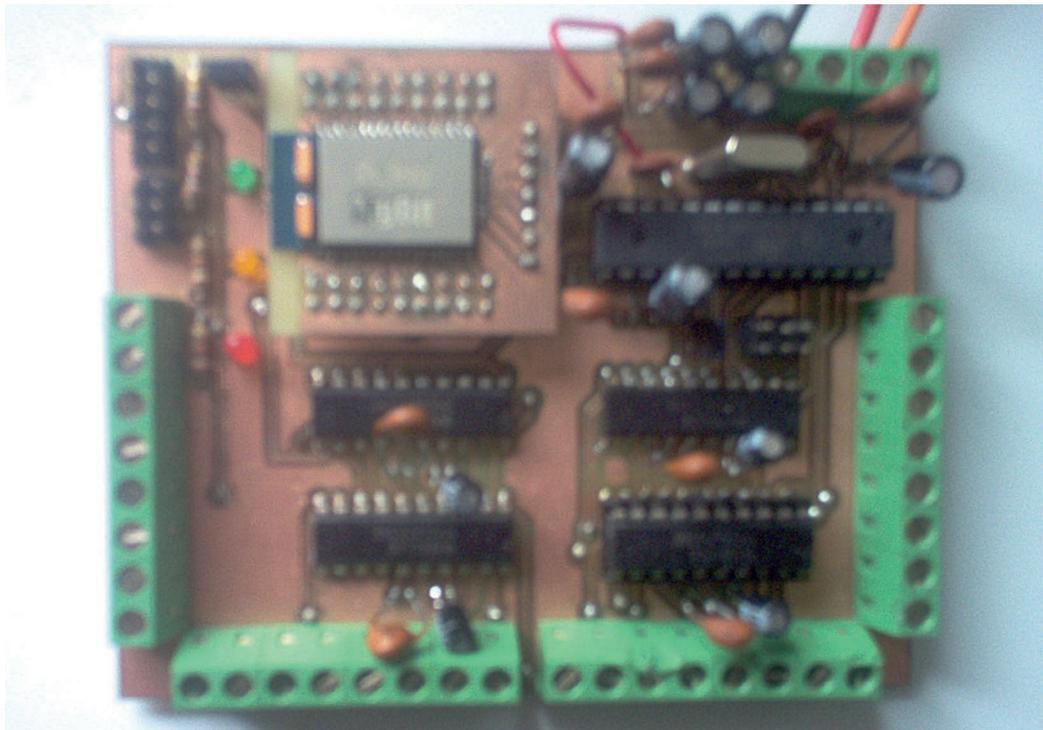


Figura 18.2. Primer prototipo del PEC.

dispositivos para placas perforadas, aumentando el tamaño y complicando la aplicación de la soldadura. Aunque el prototipo funcionó bien y ayudó con fines de prueba, estaba claro que ésta no sería la tecnología para la producción en masa.

Primera versión

Una vez que el prototipo funcionó bien y pudimos comunicarnos de forma inalámbrica entre la placa y una placa AVR Raven procedimos a fabricar la primera versión de la placa con el objetivo específico de controlar tantos motores de baja potencia como fuera posible. La placa es simple, con un módulo ZigBit, un Mega168 y un conjunto de doce registros de desplazamiento de 8 salidas TPIC6A596DW con transistores de potencia integrados en cada salida. Tener el transistor de potencia dentro del registro de desplazamiento simplificó significativamente el diseño y conseguimos 96 salidas que podían controlar motores pequeños sin demasiados equipos externos. Esta placa también tiene ocho puertos que funcionaban como entradas analógicas permitiendo detectar cambios externos.

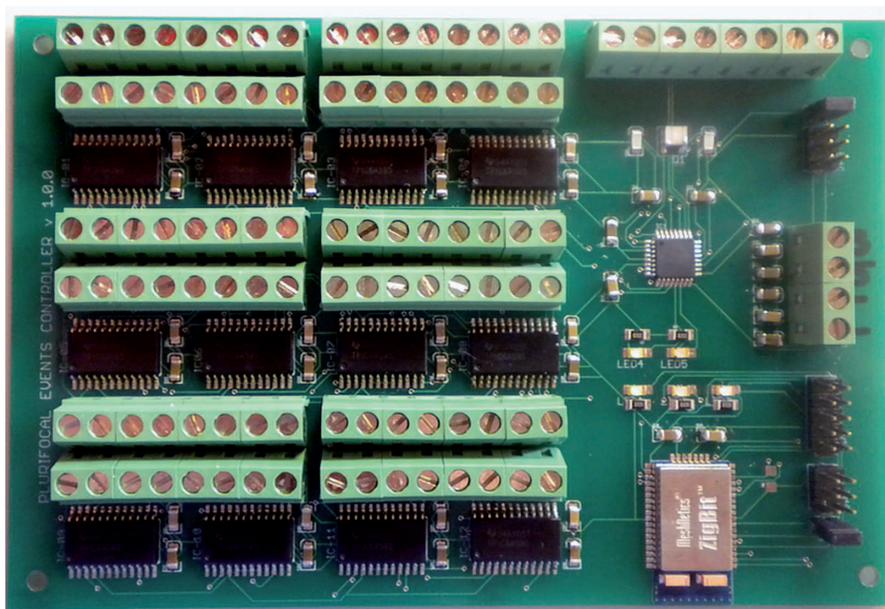


Figura 18.3. Primera versión del PEC.

Dos códigos diferentes pero interconectados tenían que ser desarrollados para el módulo: *i)* el código que maneja la comunicación ZigBee dentro del módulo ZigBit; y *ii)* el código en el microcontrolador principal que maneja los controles del motor y se comunica con el módulo ZigBee.

Con la placa física fabricada se probaron varias versiones del código. Inicialmente el módulo ZigBee utilizó la solución prefabricada SerialNet. Permitted conectarse a la red, enviar y recibir mensajes fácilmente, pero no se pudieron seleccionar algunos parámetros y funcionó de manera irregular en determinadas circunstancias. Posteriormente, la versión SerialNet fue reemplazada por un software personalizado que permitió configuraciones de red en estrella. En ese momento no había proyectos artísticos que requirieran redes de malla.

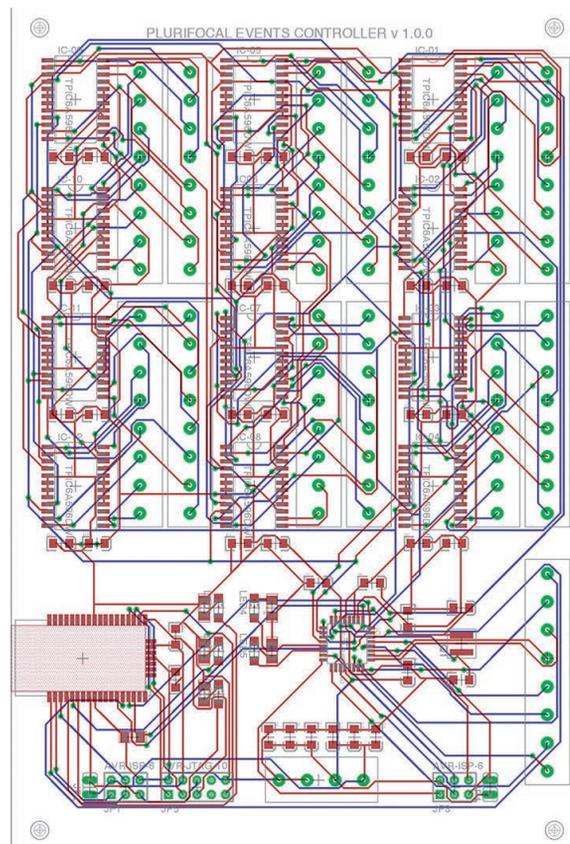


Figura 18.4. Diagrama de la primera versión del PEC.

El código del microcontrolador principal también cambió en varias ocasiones. No sólo con cambios en el código del módulo ZigBit, sino también con la exploración de las propiedades sonoras de los motores. Mientras los parámetros, rangos y protocolos cambiaron durante el tiempo, la lógica o *estado de flujo* siguió siendo similar. Al cargar el arranque, el código pide al módulo ZigBit que se conecte a la red y espere hasta que se establezca la conexión. Una vez que la placa está conectada a la red comienza a enviar mensajes frecuentes al coordinador con los valores de los ocho registros ADC. También activa una interrupción que espera mensajes procedentes del módulo ZigBit. El mensaje que proviene del coordinador representa el valor relativo que cada puerto de salida debe establecer. En esta versión el PEC sólo manejaba valores discretos de encendido/apagado para los motores.

Segunda versión

La primera versión del circuito se utilizó en toda la serie de la obra *Metáforas para pianos muertos*, y funcionó sin ningún problema. Sin embargo, al planificar nuevas obras quedó claro que era necesario actualizar dos elementos que no habían sido considerados en la

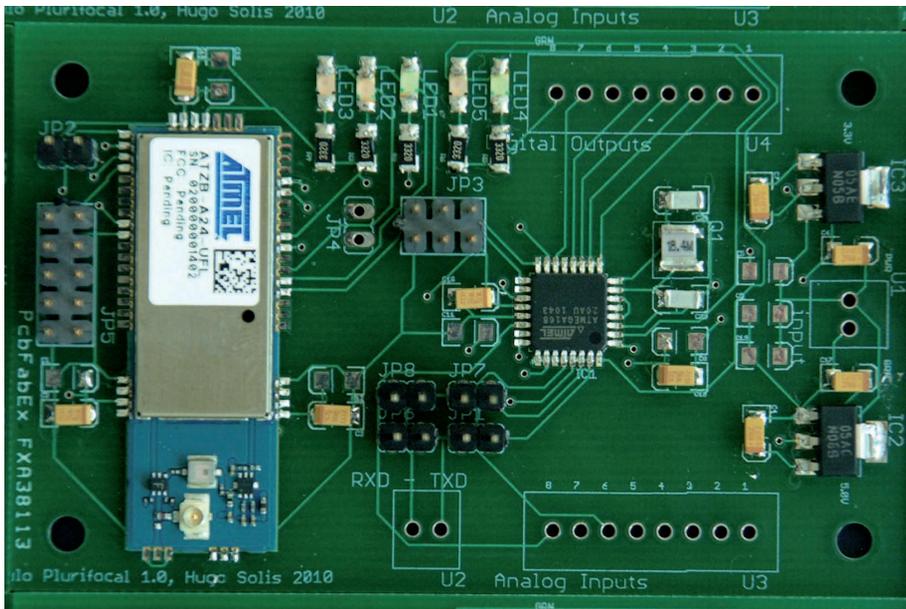


Figura 18.5. Segunda versión del PEC sin terminar.

primera versión. El primero fue el deseo de difundir los dispositivos de percusión sobre espacios más grandes, lo que significaba que necesitábamos utilizar antenas más grandes; la segunda era la posibilidad de controlar motores más grandes, lo que significaba que necesitábamos amplificadores de potencia más grandes.

Con esto en mente se diseñó y produjo una segunda versión del Plurifocal Events Controller. Con la experiencia de la placa anterior se creó un circuito modular más pequeño con una mejor configuración. La placa ahora incluye un conjunto de reguladores de tensión, lo que nos permite utilizar una variedad de baterías o reguladores externos, y ahora también utiliza el módulo de amplificador de ZigBit con la antena externa, ofreciendo una conexión mucho más larga de cerca de 100 metros.

Puesto que el objetivo de esta placa era controlar algunos dispositivos a distancias más largas en lugar de controlar muchos motores cerca uno del otro, sólo tiene ocho puertos de salida. Por otro lado, el beneficio es que con dos relojes internos y una matriz de variables la salida de cada puerto es ahora ajustable mediante la modulación de ancho de pulso (PWM). El uso de PWM en las salidas nos permitió experimentar con la fuerza y la velocidad de los solenoides y motores, dando lugar a la generación de dinámicas y cambios de amplitud en los materiales sonoros.

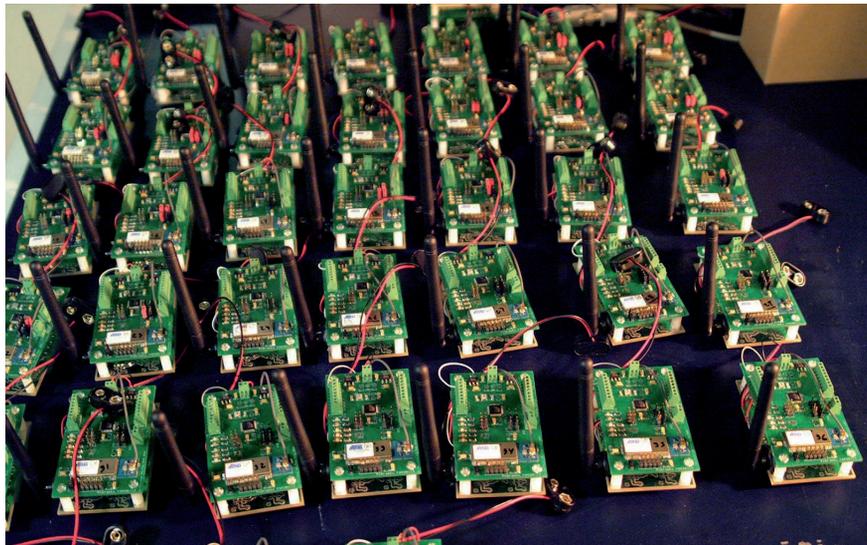


Figura 18.6. Segunda versión del PEC terminado.

La placa no tiene ningún amplificador de potencia dejando las capacidades de potencia a un circuito secundario que puede ser modificado o sustituido de acuerdo con la obra de arte. Hasta la fecha sólo se ha desarrollado una versión de la tarjeta de alimentación. Usando los transistores Darlington de potencia y los PCB de *huesos desnudos*, se fabricó a un bajo costo una solución barata de cuatro amplificadores de potencia unidireccionales y uno bidireccional. Los transistores TIP120 permiten el control de motores y solenoides de tamaño mediano y de alta corriente.

Como en el caso de la primera placa varias versiones del código se desarrollaron con el tiempo, la última versión demostró ser una solución robusta y maleable. El módulo ZigBit contiene un código C personalizado que se conecta automáticamente a la red de malla, configurando cada placa como un ruteador. Tiene un conjunto de características simples pero importantes; por ejemplo, si se pierde la conexión, el módulo intenta volver a conectarse; si se reciben mensajes dañados, se descartan, etcétera.

El código en el microcontrolador principal tuvo la mejoría más significativa. Dejar todas las funciones de conexión y radio al módulo ZigBit permite al ATmega168 centrarse en el control de solenoides y motores. Con una conexión directa entre el registro personalizado del microcontrolador y los pines de salida, se obtuvo una mejor resolución. La velocidad de control de los puertos es, en esta versión, lo suficientemente rápida como para controlar no sólo la velocidad y la fuerza sino también el pulso de repetición. El protocolo permite el direccionamiento individual, la sincronización, los parámetros globales y el cambio de modulación, de tal manera que la transmisión de datos sobre el aire se reduce en comparación con las iteraciones anteriores. La simplificación de los mensajes y la reducción de los datos transmitidos produjeron una red altamente confiable con mejor resolución, permitiendo muchos más tipos de generación de sonido y patrones rítmicos.

FABRICACIÓN DE COMPONENTES MECÁNICOS

La razón para desarrollar el sistema Plurifocal Events Controller fue producir sonidos acústicos controlados con resolución digital extendida sobre grandes territorios físicos. Sin embargo, al final del proceso, el sonido tiene que ser generado con algún tipo de activador

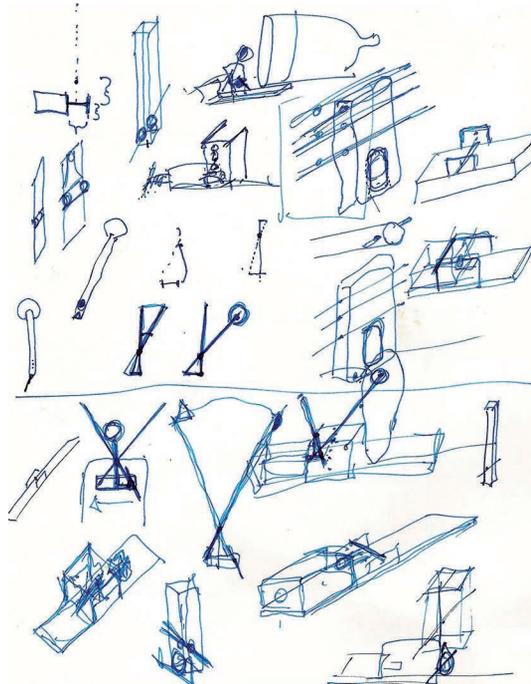


Figura 18.7. Bocetos iniciales del sistema percutor.

mecánico. Durante los años que duró el desarrollo de la placa hubo una exploración de motores y solenoides. La exploración no fue ciertamente metodológica y se basó más en preocupaciones pragmáticas y económicas.

La adquisición de nuevos motores y solenoides en grandes cantidades puede ser extremadamente costosa. Por esta razón la obtención de piezas usadas en almacenes excedentes era una metodología natural.⁴

Las primeras exploraciones se hicieron con pequeños motores de corriente continua que nos permitieron frotar cuerdas de piano. Mientras que el sonido era atractivo, la vida útil era extremadamente corta. Los motores DC de tamaño mediano más tarde se utilizaron como percutores pendulares. Tiempo después, los motores vibratorios extremadamente pequeños se utilizaron para vibrar cuerdas de piano por empatía. La exploración del uso de motores

⁴ La tienda en línea <http://allelectronics.com> es la fuente principal para los motores y los solenoides en los Estados Unidos. En Japón, el distrito de Akihabara es un lugar extraordinario para conseguir motores. En México la calle de República del Salvador, en el Centro Histórico es también un buen lugar para obtener motores usados y solenoides de impresoras y equipo electrónico reciclado.

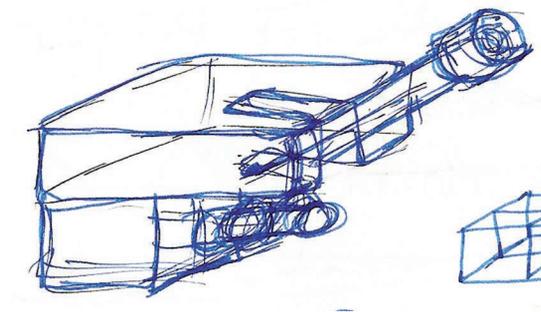


Figura 18.8. Boceto de la versión final del sistema percutor.

para golpear objetos de una manera semiaóptica pero rítmica continuó durante una residencia de arte en Japón. De regreso en México se obtuvo un conjunto de potentes solenoides y se desarrolló un mecanismo de percusión que permitió ataques fuertes y precisos.



Figura 18.9. Bases de madera para solenoides.



Figura 18.10. Solenoides de 24 volts.

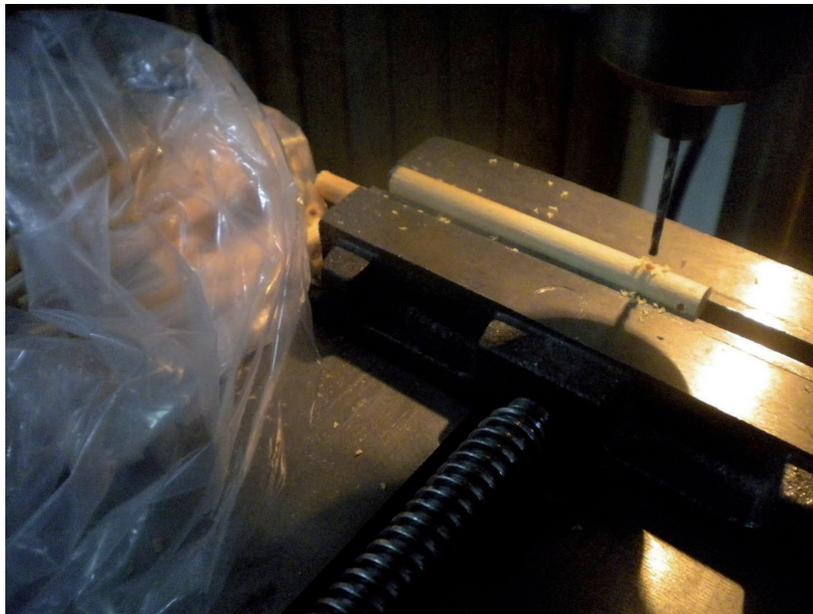


Figura 18.11. Perforación de baquetas para percutores.



Figura 18.12. Cabezas de amortiguamiento sonoro para baquetas.



Figura 18.13. Calibración y colocación de resortes para solenoides.

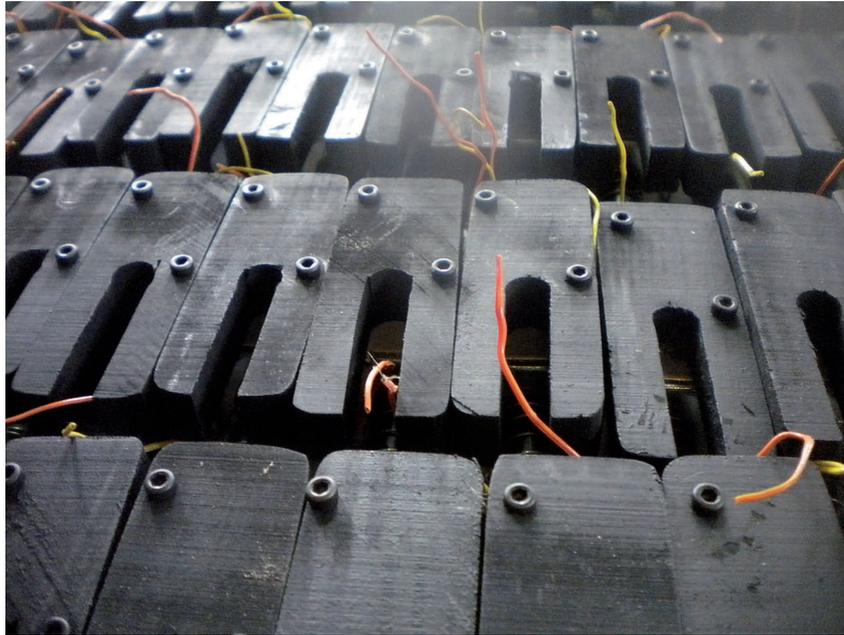


Figura 18.14. Ensamblaje de bases para solenoides.

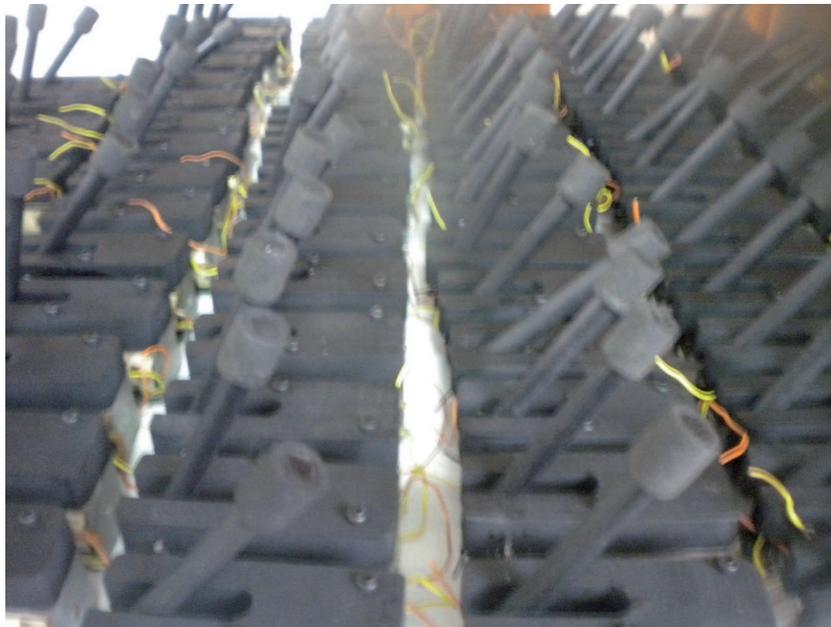


Figura 18.15. Juego de solenoides terminados.

APLICACIONES DEL PLURIFOCAL EVENTS CONTROLLER

Las diferentes iteraciones y versiones del Plurifocal Events Controller se han utilizado en varias obras de arte. Los detalles de estas obras se pueden encontrar en los otros capítulos de este libro. Sin embargo, es importante mencionar las obras que se han creado con los módulos especialmente por el lazo de retroalimentación creativa entre los objetivos artísticos y expresivos de cada obra y el desarrollo del módulo.

Metáforas para Pianos Muertos I La primera versión del módulo se empleó en este trabajo en el que varios motores de corriente continua fueron controlados por el PEC. Usando el protocolo SerialNet, un pequeño microcontrolador envió mensajes al PEC. Para este primer trabajo no hubo valores de modulación, sólo eran posibles los estados *prendido* y *apagado*. Como los motores de corriente continua consumían aproximadamente 800 miliamperios cada uno, se necesitaba una tarjeta de alimentación externa con una matriz de transistores de potencia además de los transistores integrados en el circuito principal.

Metáforas para Pianos Muertos II La segunda versión de este trabajo, que también utilizó la primera versión del circuito, requirió cuatro pianos suspendidos. El protocolo SerialNet se siguió utilizando, ahora con una configuración de red en estrella. Las placas también eran controladas por un pequeño microordenador Linux. Los motores estaban colgados de cables que caían caóticamente en la placa de sonido. Estos nuevos motores no consumían mucha corriente, por tanto, no se utilizó ninguna tarjeta de alimentación externa.

Metáforas para pianos muertos III La tercera versión de este trabajo utilizó dos pianos y también empleó la primera versión del tablero. Se utilizó una aplicación BitCloud personalizada en los módulos ZigBit y se actualizó el protocolo de comunicación. Pequeños motores vibradores se colocaron en las cuerdas simplificando el diseño de la obra y ofreciendo una paleta de sonidos mucho más rica. El microordenador fue reemplazado por una computadora regular y la tarjeta coordinadora fue conectada por USB.

Ensemble Modern Concert La segunda versión del Plurifocal Events Controller se estrenó con un concierto con el Ensemble Modern. Doce PEC se distribuyeron por toda

una sala de conciertos. Cada PEC activaba motores que golpeaban las paredes metálicas del lugar. SerialNet se utilizó de nuevo, pero ahora había modulación de pulsos en las salidas y con ello control de dinámica. El uso de una topología de malla permitió una conexión a distancias más largas.

Concierto para piano, viola y galería También utilizando la nueva versión del PEC, las paredes de la galería fueron golpeadas de una manera similar a la obra anterior. Se probaron cambios menores en los protocolos y el código.

Ecos para locaciones olvidadas en el Laboratorio Arte Alameda La segunda versión de la placa se utilizó para controlar 96 solenoides de alta potencia. El protocolo de transmisión se actualizó, se desarrollaron los parámetros de sincronización, pero volvimos a la condición no modulada de *prendido y apagado*.

Ecos para locaciones olvidadas para el festival Transitio_MX Esta pieza empleó una configuración física similar al trabajo anterior, pero con software actualizado que permitió una gran variedad de texturas sónicas usando modulación de pulsos.

Axial Para este trabajo toda la experiencia anterior contribuyó al logro de una tecnología robusta y eficiente. El protocolo de datos permitió la modulación no sólo de volumen sino también de estructuras rítmicas. Una colección de mensajes para la sincronización y los cambios globales permitió juegos rítmicos. Los pulsos rítmicos ahora se generan dentro del circuito, si es necesario, reduciendo significativamente los datos transmitidos entre los módulos, que ahora están en una red de la malla de módulos ruteadores, lo que permite las conexiones de varios cientos de metros entre módulos. Un código BitCloud personalizado reemplazó a SerialNet creando una red inalámbrica mucho más robusta.

MATERIALES

La tabla 18.6 enumera las partes necesarias para construir un módulo. La tabla 18.1. enumera las partes necesarias para construir el módulo de potencia.⁵

⁵ No se recomienda replicar el módulo de potencia ya que hay módulos más simples y económicos para impulsar motores y solenoides.

Tabla 18.1

<i>Cantidad</i>	<i>Descripción</i>	<i>Número de parte de Digi-key</i>	<i>Precio unitario (Dls)</i>
01	IC AVR MCU 16K 20MHZ 32TQFP	ATMEGA168-20AU-ND	\$ 2.71
01	ZigBit Module	ATZB-A24-UFLRDKR-ND	\$ 30.70
01	RESONATOR CER 18.43MHZ SMD	535-10026-1-ND	\$ 0.34
02	LED RED FACE UP 1206	P11141CT-ND	\$ 0.28
01	LED ORANGE FACE UP 1206	P11526CT-ND	\$ 0.32
02	LED GREEN FACE UP 1206	P11529CT-ND	\$ 0.29
09	CAP CERAMIC 22PF 50V NP0 1206	311-1154-1-ND	\$ 0.07
09	CAP CER 1.0UF 16V 10% X7R 1206	399-3678-1-ND	\$ 0.13
09	CAP CER 10UF 16V X7R 1206	478-1655-1-ND	\$ 0.35
01	CAP CER 100UF 6.3V X5R 1206	718-1740-1-ND	\$ 0.76
05	RES 332 OHM 1/8W .1% 1206 SMD	RNCF32T2332BICT-ND	\$ 0.20
02	TERM BLOCK 2POS SIDE ENT 2.54MM	A98333-ND	\$ 0.53
02	TERM BLOCK 8POS SIDE ENT 2.54MM	A98338-ND	\$ 2.16
01	IC REG LDO 800MA 5.0V SOT-223	LM1117IMP-5.0CT-ND	\$ 1.17
01	IC REG LDO 800MA 3.3V SOT-223	LM1117IMP-3.3CT-ND	\$ 1.17
01	ANTENNA RUBB DUCK 2.4GHZ 6 CABLE	730-1004-ND	\$ 13.84
02	CONN HEADER 12POS .100 STR 30AU	609-3363-ND	\$ 0.25

<i>Cantidad</i>	<i>Descripción</i>	<i>Número de parte de Digi-key</i>	<i>Precio unitario (Dls)</i>
08	TRANS NPN DARL 60V 5A TO-220	TIP120-ND	\$ 0.37
08	DIODE GEN PURPOSE 50V 1A DO41	1N4001FSCT-ND	\$ 0.06
08	Resistance	NA	NA
01	TERM BLOCK 2POS SIDE ENT 2.54MM	A98333-ND	\$ 0.53
01	TERM BLOCK 6POS SIDE ENT 2.54MM	A98337-ND	\$ 1.76
01	TERM BLOCK 2POS SIDE ENT 5.0MM	A98076-ND	\$ 0.39
01	TERM BLOCK 8POS SIDE ENT 5.0MM	A98080-ND	\$ 1.45

DIAGRAMAS

Los cuatro gráficos siguientes muestran los esquemas y diagramas de la versión final del Plurifocal Events Controller y del módulo de potencia. Ambas tablas fueron diseñadas usan-

do la versión freeware del software Eagle (<http://www.cadsoftusa.com>) y son de cuatro por seis pulgadas. El cableado de cada placa requería varias horas de prueba y error para generar las rutas óptimas de dos capas. Las placas fueron fabricadas por la empresa PCFabExpress (<http://www.pcfabexpress.com>). Los archivos digitales se pueden descargar de la página web del artista y son accesible para la comunidad.

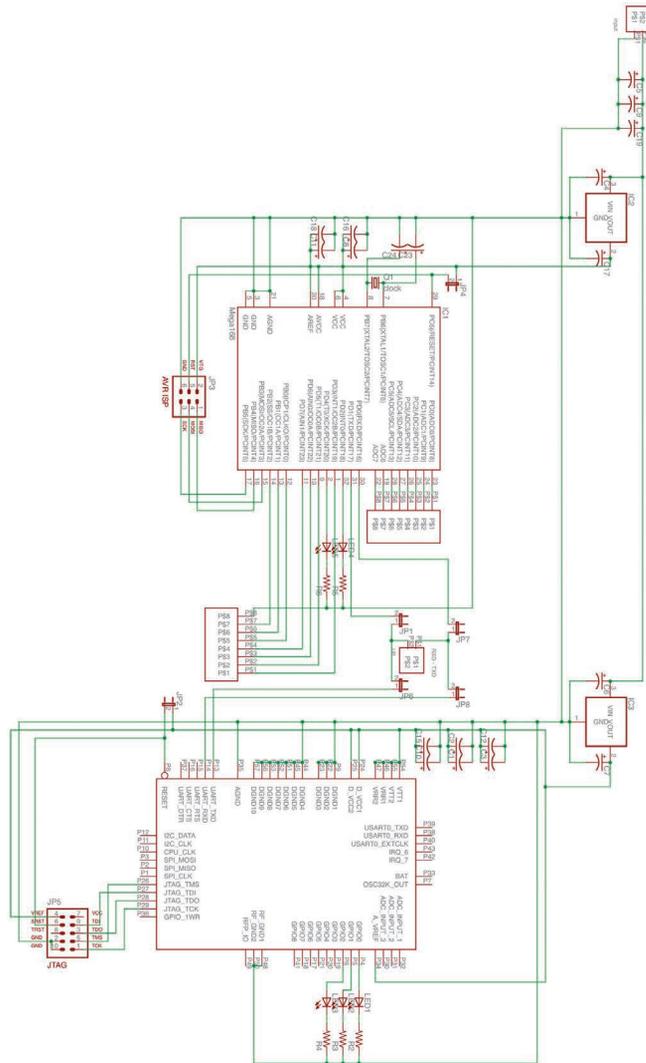


Figura 18.16. Diagrama del circuito PEC.

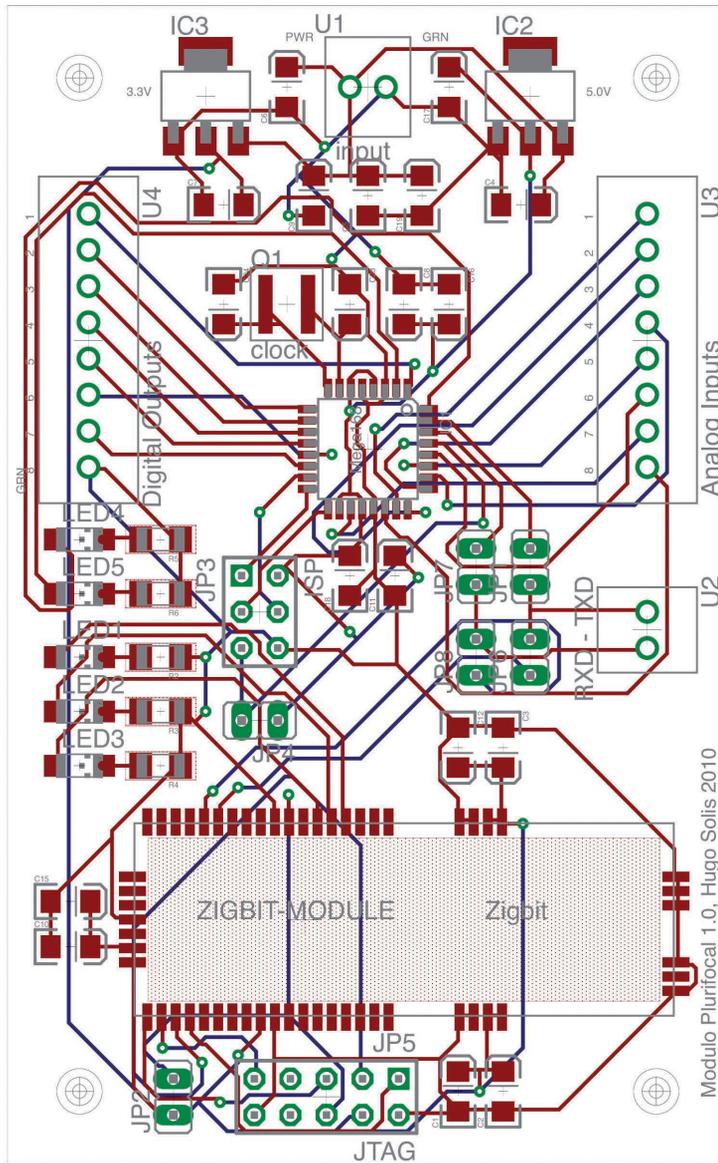


Figura 18.17. Circuito PEC.

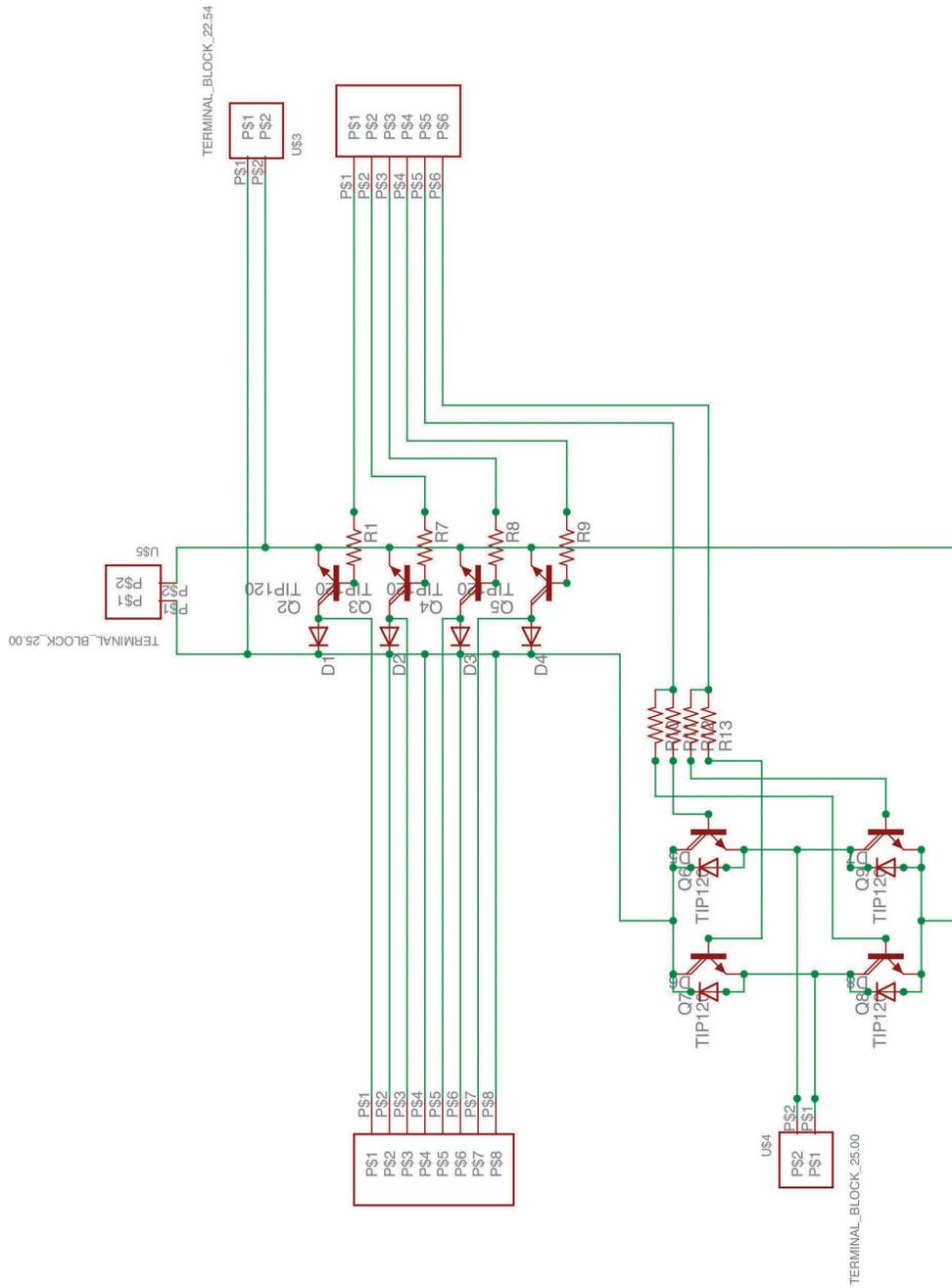


Figura 18.18. Diagrama del circuito del módulo de poder.

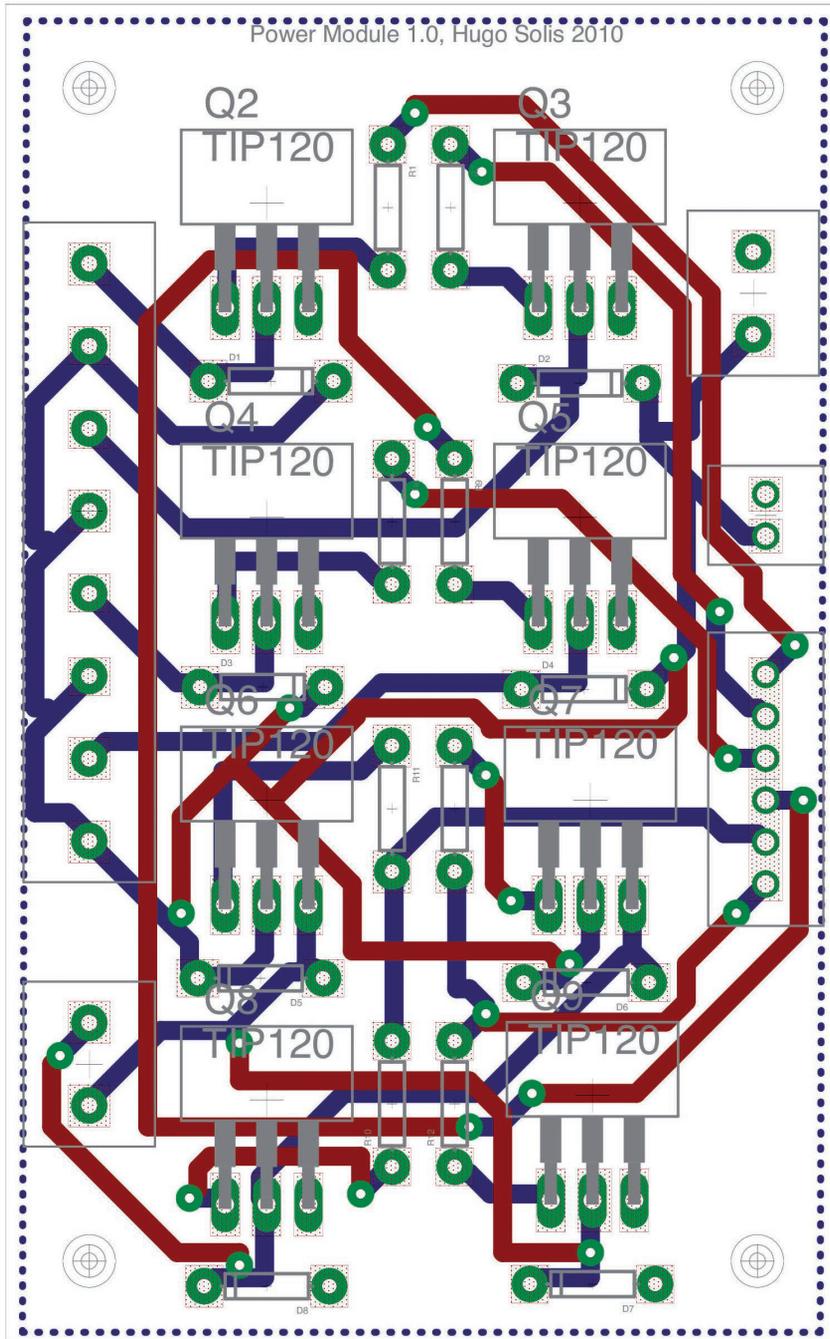


Figura 18.19. Circuito del módulo de poder.

PROTOCOLO

Protocolo ZigBee

El protocolo ZigBee cargado en el módulo ZigBee es una pequeña variación del código Peer2Peer incluido como ejemplo de la API para el paquete BitCloud proporcionado por ATMEL. Mientras que el ejemplo original de Peer2Peer proporciona una comunicación en serie entre un dispositivo maestro y un dispositivo esclavo, la modificación permite la propagación de mensajes entre todos los dispositivos esclavos y luego la extracción del identificador de destino como un mecanismo de filtrado. Esta modificación permite a un dispositivo maestro enviar mensajes a esclavos dirigidos y también recibir mensajes de todos los esclavos. El código fuente se puede descargar de la página del artista.

Rhythm Protocol

Los datos transmitidos a través de la conexión inalámbrica son muy pequeños y breves, permitiendo una transmisión de datos rápida y segura. Los mensajes se transmiten línea por línea formateados como cadena de mensajes. Cada sección del mensaje está separada por un espacio en blanco. El código fuente se puede descargar de la página del artista.

<i>Palabra</i>	<i>Descripción</i>
xx	xx es un número no utilizado en el protocolo pero transmitido por el protocolo de Bitcloud.
yy	yy es un número hexadecimal que representa el ID del módulo. Si el mensaje es 0xFF el mensaje es transmitido a todos los módulos.
zz	zz es un número hexadecimal que representa el número de comando.

<i>Comando</i>	<i>Descripción</i>
00 AA BB	ALL OUTPUTS SET INTENSITY AND PULSE. El comando 00 tiene dos partes. AA representa la intensidad como un número hexadecimal de dos dígitos. BB establece el periodo entre pulsos. Este mensaje restablece la fase inicial del pulso.
01 AA BB CC	ONE MOTOR SET INTENSITY AND PULSE. El comando 01 tiene tres partes. AA representa el motor a ser modificado. BB establece la intensidad como un número hexadecimal de dos dígitos. CC ajusta el periodo entre pulsos. Este mensaje restablece la fase de arranque del impulso del motor modificado.

02 AA BB	UN MOTOR SET INTENSITY. El comando 02 tiene dos partes. aa representa el motor a ser modificado. BB establece la intensidad como un número hexadecimal de dos dígitos. Este mensaje no restablece la fase de arranque del impulso del motor modificado.
03 AA BB	UN MOTOR SET PULSE. El comando 03 tiene dos partes. AA representa el motor a ser modificado. BB establece el pulso como un número hexadecimal de dos dígitos. Este mensaje restablece la fase inicial del impulso del motor modificado pero no la intensidad.
87 04	RESETS PULSE PHASE. El comando 04 no tiene más palabras. Este mensaje restablece la fase de impulso de todos los motores permitiendo resincronizar patrones rítmicos.
22 05	OFF PARAMETERS. El comando 05 no tiene más palabras. Este mensaje ajusta la intensidad a cero y detiene los patrones de impulsos ajustando todos los motores a apagado.

Comando	Descripción
67 01 00 30 F0	Este mensaje establecería todos los motores del módulo 01 a la intensidad de 48 (0x30) y el impulso de 240 (0xF0).
87 FF 04	Este mensaje restablecería la fase de todos los motores de todos los módulos.
45 09 01 04 FF 90	Este mensaje establecería el ID del motor 4 del módulo ID 09 a la intensidad del valor 255 (0xFF) y el impulso del 144 (0x90).
23 10 02 03 09	Este mensaje establecería la identificación del motor 3 del módulo ID 10 a la intensidad del valor 09.
78 00 05	Este mensaje desactivaría el módulo 05.
99 FF 03 02 10	Este mensaje establecería el pulso del motor id 02 de todos los módulos al valor 10.

CÓDIGO

El siguiente código presenta las instrucciones cargadas en el microcontrolador ATmega168 por los módulos Plurifocal. Este código se creó como proyecto en AVR Studio IDE. Se puede observar que el código está leyendo datos en serie usando un interruptor. Una vez que el mensaje está completo, es interpretado por la rutina principal que lo utiliza para modificar dos procesos diferentes de Modulación de Ancho de Pulso (PWM). Un PWM se utiliza para controlar la intensidad y el otro, el pulso rítmico. Para tener estos dos PWM hay que configurar dos temporizadores en la función *init*. La función principal interpreta los datos en serie utilizando el protocolo descrito en Rhythm Protocol. El código fuente del archivo c y el archivo h se pueden descargar de la página del artista (<http://HugoSolis.net>). ■ ■ ■

```

1. #include <avr/io.h>
2. #include <avr/interrupt.h>
3. #include <inttypes.h>
4. #define F_CPU 1843200UL//has a clock speed of 18.4MGH
5. #include <util/delay.h>//uses F_CPU to provide delay routines
6. #include <string.h>
7. #include <stdlib.h>
8. #include <stdio.h>
9.
10. #define ID 38
11. #define SOLENOID_SLOWER 10//for gettig 7200 per second
12. #define EVENT_UNIT 512UL
13. #define BUFFER_SIZE 32
14. #define BAUD 38400
15. #define MYUBRR F_CPU/16/BAUD-1
16. //enable output pins
17. #define SET_LED_PINS DDRD|=(1<<DDD2)|(1<<DDD4)
18. #define LED_RED (1<<PD4)
19. #define LED_RED_ON PORTD |= LED_RED
20. #define LED_RED_OFF PORTD &= ~LED_RED
21. #define LED_RED_TOGGLE PORTD ^= LED_RED
22. #define LED_GREEN (1<<PD2)
23. #define LED_GREEN_ON PORTD |= LED_GREEN
24. #define LED_GREEN_OFF PORTD &= ~LED_GREEN
25. #define LED_GREEN_TOGGLE PORTD ^= LED_GREEN
26. //enable output pins
27. #define SET_OUTPUT_PINS_D DDRD|=
28.     (1<<DDD3)|(1<<DDD5)|(1<<DDD6)|(1<<DDD7)
29. //enable output pins
30. #define SET_OUTPUT_PINS_B DDRB|=(1<<DDB0)|(1<<DDB1)|(1<<DDB2)
31.
32. #define OUTPUTS 7
33. #define OUTPUT_1 (1<<PD3)
34. #define OUTPUT_2 (1<<PD5)
35. #define OUTPUT_3 (1<<PD6)
36. #define OUTPUT_4 (1<<PD7)
37. #define OUTPUT_5 (1<<PB0)
38. #define OUTPUT_6 (1<<PB1)
39. #define OUTPUT_7 (1<<PB2)
40.
41. #define OUTPUT_1_ON PORTD |= OUTPUT_1
42. #define OUTPUT_1_OFF PORTD &= ~OUTPUT_1
43. #define OUTPUT_2_ON PORTD |= OUTPUT_2
44. #define OUTPUT_2_OFF PORTD &= ~OUTPUT_2
45. #define OUTPUT_3_ON PORTD |= OUTPUT_3
46. #define OUTPUT_3_OFF PORTD &= ~OUTPUT_3
47. #define OUTPUT_4_ON PORTD |= OUTPUT_4
48. #define OUTPUT_4_OFF PORTD &= ~OUTPUT_4
49. #define OUTPUT_5_ON PORTB |= OUTPUT_5
50. #define OUTPUT_5_OFF PORTB &= ~OUTPUT_5
51. #define OUTPUT_6_ON PORTB |= OUTPUT_6

```

```

52. #define OUTPUT_6_OFF PORTB &= ~OUTPUT_6
53. #define OUTPUT_7_ON PORTB |= OUTPUT_7
54. #define OUTPUT_7_OFF PORTB &= ~OUTPUT_7
55.
56. typedef enum
57. {
58.     NEW_MESSAGE_YES, //There is a new message
59.     NEW_MESSAGE_NO //There is not a new message
60. } NewMessageState_t;
61.
62. typedef enum
63. {
64.     SOLENOID_INTERRUPT_DIRTY_STATE, //The values has to be updated
65.     SOLENOID_INTERRUPT_UPDATED_STATE //The values hare updated
66. } SolenoidInterruptState_t;
67.
68. typedef enum
69. {
70.     EVENT_READY,
71.     EVENT_DONE,
72. } EventState_t;

```

```

1. #include "plurifocal.h"
2. /*****
3. *****/
4. DEFINITION OF FUNCTIONS
5. *****/
6. *****/
7. void init(void);
8. void usart_init(void);
9. void usart_putc(char c);
10. void usart_puts(char *s);
11.
12. /*****
13. *****/
14. DEFINITION OF GLOBAL VARIABLES
15. *****/
16. *****/
17. volatile static NewMessageState_t newMessageState =
18.     NEW_MESSAGE_NO;
19. static char messageBuffer[BUFFER_SIZE];
20. static char message[BUFFER_SIZE];
21. volatile static uint8_t messagePosition;
22. volatile static uint8_t messageSize;
23.
24. volatile static SolenoidInterruptState_t
25.     solenoidInterruptState = SOLENOID_INTERRUPT_UPDATED_STATE;
26.
27. volatile static uint16_t port_period[OUTPUTS];

```

```

28. volatile static uint16_t port_period_counter[OUTPUTS];
29. volatile static uint8_t port_counter[OUTPUTS];
30. volatile static uint8_t port_intensity[OUTPUTS];
31.
32. volatile static EventState_t eventState = EVENT_READY;
33.
34. /*****
35. *****/
36. MAIN
37. *****/
38. *****/
39. int main(void){
40.     _init();
41.     usart_init();
42.     sei();
43.
44.     while(1){
45.         //HANDLE NEW INCOMING MESSAGES
46.         if(newMessageState == NEW_MESSAGE_YES){
47.             newMessageState = NEW_MESSAGE_NO;
48.
49.             strtok(message, " "); //get ridd of first number
50.
51.             char *idString = strtok(NULL, " "); //get first number
52.             uint16_t id = strtol(idString, NULL, 16); //convert to int
53.             if(id == ID || id == 0xFF){
54.                 LED_GREEN_ON;
55.
56.                 char *commandString = strtok(NULL, " "); //get first number
57.                 uint16_t command = strtol(commandString, NULL, 16);
58.
59.                 //all motors intensiy pulse in SYNK
60.                 if(command == 00){
61.                     char *intensityString = strtok(NULL, " ");
62.                     uint16_t intensity = strtol(intensityString, NULL, 16);
63.                     char *periodString = strtok(NULL, " "); //get first number
64.                     uint16_t period = strtol(periodString, NULL, 16);
65.
66.                     for(int i = 0; i < OUTPUTS; i++){
67.                         port_period[i] = period;
68.                         port_intensity[i] = intensity;
69.                         port_counter[i] = 0;
70.                         port_period_counter[i] = 0;
71.                     }
72.                 }
73.
74.                 //one motor intensity and pulse
75.                 else if(command == 01){
76.                     char *motorString = strtok(NULL, " "); //get first number
77.                     uint16_t motor = strtol(motorString, NULL, 16);
78.                     char *intensityString = strtok(NULL, " ");

```

```
79.  uint16_t intensity = strtol(intensityString, NULL, 16);
80.  char *periodString = strtok(NULL, " "); //get first number
81.  uint16_t period = strtol(periodString, NULL, 16);
82.
83.  port_period[motor] = period;
84.  port_intensity[motor] = intensity;
85.  port_counter[motor] = 0;
86.  port_period_counter[motor] = 0;
87.  }
88.
89.  //one motor intensity
90.  else if(command == 02){
91.  char *motorString = strtok(NULL, " "); //get first number
92.  uint16_t motor = strtol(motorString, NULL, 16);
93.  char *intensityString = strtok(NULL, " ");
94.  uint16_t intensity = strtol(intensityString, NULL, 16);
95.
96.  port_intensity[motor] = intensity;
97.  }
98.
99.
100. //one motor pulse
101. else if(command == 03){
102. char *motorString = strtok(NULL, " "); //get first number
103. uint16_t motor = strtol(motorString, NULL, 16);
104. char *periodString = strtok(NULL, " "); //get first number
105. uint16_t period = strtol(periodString, NULL, 16);
106.
107. port_period[motor] = period;
108. port_period_counter[motor] = 0;
109. }
110.
111.
112. //SYNCK
113. if(command == 04){
114. for(int i = 0; i < OUTPUTS; i++){
115.     port_counter[i] = 0;
116.     port_period_counter[i] = 0;
117. }
118. }
119.
120. //OFF
121. if(command == 05){
122. for(int i = 0; i < OUTPUTS; i++){
123.     port_period[i] = 0;
124.     port_intensity[i] = 0;
125.     port_counter[i] = 0;
126.     port_period_counter[i] = 0;
127. }
128. }
129. }
```

```
130. }
131.
132. //SOLENOID CONTROLS
133. if(solenoidInterruptState == SOLENOID_INTERRUPT_DIRTY_STATE){
134.     solenoidInterruptState = SOLENOID_INTERRUPT_UPDATED_STATE;
135.
136. //CONTROL OF OUTPUT 1
137. if(port_counter[0] == 0){
138.     OUTPUT_1_ON;
139.     LED_RED_ON;
140. }
141. if(port_counter[0] < port_intensity[0]) port_counter[0]++;
142. if(port_counter[0] == port_intensity[0]){
143.     OUTPUT_1_OFF;
144.     LED_RED_OFF;
145. }
146.
147. //CONTROL OF OUTPUT 2
148. if(port_counter[1] == 0){
149.     OUTPUT_2_ON;
150.     LED_RED_ON;
151. }
152. if(port_counter[1] < port_intensity[1]) port_counter[1]++;
153. if(port_counter[1] == port_intensity[1]){
154.     OUTPUT_2_OFF;
155.     LED_RED_OFF;
156. }
157.
158. //CONTROL OF OUTPUT 3
159. if(port_counter[2] == 0){
160.     OUTPUT_3_ON;
161.     LED_RED_ON;
162. }
163. if(port_counter[2] < port_intensity[2]) port_counter[2]++;
164. if(port_counter[2] == port_intensity[2]){
165.     OUTPUT_3_OFF;
166.     LED_RED_OFF;
167. }
168.
169. //CONTROL OF OUTPUT 4
170. if(port_counter[3] == 0){
171.     OUTPUT_4_ON;
172.     LED_RED_ON;
173. }
174. if(port_counter[3] < port_intensity[3]) port_counter[3]++;
175. if(port_counter[3] == port_intensity[3]){
176.     OUTPUT_4_OFF;
177.     LED_RED_OFF;
178. }
179.
180. //CONTROL OF OUTPUT 5
```

```
181. if(port_counter[4] == 0){
182.   OUTPUT_5_ON;
183.   LED_RED_ON;
184. }
185. if(port_counter[4] < port_intensity[4]) port_counter[4]++;
186. if(port_counter[4] == port_intensity[4]){
187.   OUTPUT_5_OFF;
188.   LED_RED_OFF;
189. }
190.
191. //CONTROL OF OUTPUT 6
192. if(port_counter[5] == 0){
193.   OUTPUT_6_ON;
194.   LED_RED_ON;
195. }
196. if(port_counter[5] < port_intensity[5]) port_counter[5]++;
197. if(port_counter[5] == port_intensity[5]){
198.   OUTPUT_6_OFF;
199.   LED_RED_OFF;
200. }
201.
202. //CONTROL OF OUTPUT 7
203. if(port_counter[6] == 0){
204.   OUTPUT_7_ON;
205.   LED_RED_ON;
206. }
207. if(port_counter[6] < port_intensity[6]) port_counter[6]++;
208. if(port_counter[6] == port_intensity[6]){
209.   OUTPUT_7_OFF;
210.   LED_RED_OFF;
211. }
212. }
213.
214.
215. //CONTROL OF PULSE
216. if(eventState == EVENT_READY){
217.   LED_GREEN_OFF;
218.   eventState = EVENT_DONE;
219.   for(int i = 0; i < OUTPUTS; i++){
220.     if(port_period[i] != 0){
221.       port_period_counter[i]++;
222.       if(port_period_counter[i] == port_period[i]){
223.         port_period_counter[i] = 1;
224.         port_counter[i] = 0;
225.       }
226.     }
227.   }
228. }
229. }
230. return 0;
231. }
```

```

232.
233.
234.      /*****
235. *****/
236. FUNCTIONS FOR CONTROL
237. *****/
238. *****/
239.      void _init(void){
240.  for(int i = 0; i < OUTPUTS; i++){
241.  port_period[i] = 0;
242.  port_intensity[i] = 0;
243.  port_counter[i] = 0;
244.  port_period_counter[i] = 0;
245.  }
246.
247.  srand(ID);
248.  _delay_ms(250); // wait to clock stablization
249.
250.  //DEFINING THE PINS FOR LEDS
251.  SET_LED_PINS;
252.
253.  //turn the leds off
254.  LED_RED_OFF;
255.  LED_GREEN_OFF;
256.
257.  //DEFINIE THE PINS FOR DIGITAL OUTPUTS
258.  SET_OUTPUT_PINS_B;
259.  SET_OUTPUT_PINS_D;
260.
261.  //put outputs off
262.  OUTPUT_1_OFF;
263.  OUTPUT_2_OFF;
264.  OUTPUT_3_OFF;
265.  OUTPUT_4_OFF;
266.  OUTPUT_5_OFF;
267.  OUTPUT_6_OFF;
268.  OUTPUT_7_OFF;
269.
270.
271.  //TIMER0 FOR SOLENOID RESOLUTION TIMES
272.  //setting the compare match overflow A
273.  TIMSK0 |= (1 << OCIE0A);
274.  TCCR0B |= (4 << CS00); //prescalar 256
275.  OCR0A = SOLENOID_SLOWER;
276.
277.
278.  //TIMER1 FOR RYTHM
279.  //setting the compare match overflow A
280.  TIMSK1 |= (1 << OCIE1A);
281.  TCCR1B |= (5 << CS10); //prescalar 1024
282.  OCR1A = EVENT_UNIT;

```

```

283.     }
284.
285.     ISR(TIMER0_COMPA_vect){
286. TCNT0 = 0;
287. solenoidInterruptState = SOLENOID_INTERRUPT_DIRTY_STATE;
288.     }
289.
290.     ISR(TIMER1_COMPA_vect){
291. TCNT1 = 0;
292. eventState = EVENT_READY;
293.     }
294.
295. /*****
296. *****/
297. FUNCTIONS FOR THE SERIAL CONTROL OF THE WILRESMODULE
298. *****/
299. *****/
300.     void usart_init(void)
301.     {
302. /*Set baud rate */
303. unsigned int ubrr = MYUBRR;
304. UBRR0H = (unsigned char)(ubrr>>8);
305. UBRR0L = (unsigned char)ubrr;
306. /*Enable receiver and transmitter and receiver interrupt */
307. UCSR0B |= (1<<RXEN0)|(1<<TXEN0)|(1<<RXCIE0);
308. /* Set frame format: 8data, 1 stop parity */
309. //DO NOTHING BECUASE WE USE DEFAULT VALUES
310.     }
311.
312.     void usart_putc(char c) {
313. // wait until UDR ready
314. while(!(UCSR0A & (1 << UDRE0)));
315. UDR0 = c; // send character
316.     }
317.
318.     void usart_puts(char *s) {
319. // loop until *s != NULL
320. while (*s) {
321. usart_putc(*s);
322. s++;
323. }
324.     }
325.
326.
327.     //USART receive
328.     ISR(USART_RX_vect){
329.
330. char c = UDR0;
331. if(c != '\r' || c != '\n'){
332. if(c == 'N'){
333. memcpy(message, messageBuffer, messagePosition);

```

```
334. messageSize = messagePosition;
335. messagePosition = 0;
336. newMessageState = NEW_MESSAGE_YES;
337. }
338. else memcpy(messageBuffer + messagePosition++, &c, 1);
339. }
340. }
```


Conclusión

En este libro se hace un breve recorrido sobre la producción artística que he desarrollado desde poco antes de terminar los estudios de la licenciatura en Piano, hasta poco después de haber terminado los estudios de doctorado en Arte Digital y Medios Experimentales. La intención de este libro ha sido mostrar un proceso, o más bien, los procesos y evoluciones a lo largo de estos años con la finalidad de mostrar –sobre todo a los estudiantes de arte electrónico–, tanto aquello que hay detrás de la creación de una obra particular, como la evolución en el conjunto de las obras vistas como un todo.

Realizar este recorrido ha sido todo un ejercicio pedagógico para mí mismo. Me ha permitido revisar el trabajo pasado, evaluar sus cualidades y contradicciones y revisar las relaciones entre todas las obras. Sin duda, como se mencionó en la introducción, hay trabajo que a mi juicio es valioso y también trabajo que podría o debería haber sido de otro modo. Sin embargo, el pasado es inamovible y, como se dice “lo hecho hecho está”. No hay manera de dar marcha atrás para subsanar los errores pero sí –y esa es la intención de este libro– de evaluar lo logrado y lo mal logrado y con ello tener herramientas para que el trabajo futuro sea una mejora y una evolución acertada.

Realizar esta revisión de trabajos es además una manera de recorrer el momento histórico en que han sido producidas. Las primeras piezas fueron producidas antes del establecimiento de los programas de estudio que ahora existen y antes de la asimilación, por parte de la comunidad, de herramientas como Processing o Arduino. Los primeros desarrollos

tecnológicos presentados en este libro son previos al establecimiento de internet en México. Con esto quiero decir que parte del trabajo presentado en este texto requirió indagar en libros de manera solitaria para solucionar problemas simples. Este libro se termina cuando los foros, espacios de ayuda, colectivos y espacios de internet son más que abundantes. Otra conclusión, tal vez obvia, que se podría obtener de este texto es que en los 15 años que abarca, las mecánicas de trabajo han cambiado significativamente y muy probablemente para bien. Los creadores de arte electrónico tienen ahora a su disposición un torrente de información, plataformas y ejemplos que les permiten concentrarse en los procesos creativos y expresivos en lugar de resolver problemas de implementación triviales.

Por lo contrario –y esta sería otra conclusión– es que un elemento que a mi juicio sería fundamental enfatizar es mi inclinación al dominio de la técnica. Los creadores de arte electrónico deben dominar sus técnicas de trabajo como se da en cualquier otro espacio de trabajo. Los productores de arte electrónico deben poder programar, realizar sistemas electrónicos y mecánicos y entender los procesos que se dan en la electrónica analógica y digital independientemente del territorio en el que realicen su labor. La profundidad en el dominio de estas herramientas es libertad de cada individuo, pero durante estos años de dialogar y convivir con profesionales, gurús y personas en formación me es cada día más evidente que contar con estas herramientas abre horizontes creativos invaluableles.

Ahora bien, el verdadero objetivo no es tanto que esta revisión me sirva a mí como creador para entender los procesos, sino que les sirva a los estudiantes de arte electrónico para que, si es el caso, puedan revisar ejemplos de procedimientos, técnicas, procesos y mecánicas de desarrollo que podría replicar en su trabajo.

El objetivo no es mostrar los procesos de trabajo para su réplica o repetición sino más bien mostrar los entramados para que sirvan, en el mejor de los casos, de inspiración o motivación para la creación de obras originales.

Por otra parte, otra lectura que me gustaría que los estudiantes logaran es en relación a los tiempos en los procesos de producción. Las obras requieren tiempo de producción y en ésta es importante incluir factores muy importantes entre los que se encuentran los procesos de aprendizaje, los procesos de asimilación de estéticas y conocimientos, la apropiación de recursos, la adquisición de habilidades, la búsqueda de espacios para la producción, la

obtención de recursos económicos, la obtención de espacios de exhibición y finalmente la creación de públicos interesados en las propuestas.

Completar esta cadena de elementos requiere un esfuerzo y una constancia que sólo pueden obtenerse a través de una voluntad muy profunda y por una necesidad creativa y expresiva auténtica. Sin embargo, si existe la voluntad de producir obras, sin duda que es posible. Los recursos existen y es cuestión de tener la paciencia y la perseverancia para buscarlos y aprovecharlos. En mi experiencia la comunidad relacionada con el arte electrónico en México es generosa y compartida.

Sobre el autor

Hugo Solís García (México, 1976) es un artista electrónico enfocado en las posibilidades creativas y estéticas producto de la intersección entre sonido, tecnologías digitales e interactividad. Ha exhibido obras y ofrecido conciertos en México, Japón, Irlanda, Estados Unidos, España, Brasil, Italia y Portugal. Participa regularmente en actividades especializadas en el campo del arte, la ciencia y la tecnología. Presenta instalaciones sonoras, interactivas y conciertos de manera regular.

Ha recibido apoyos, becas y reconocimientos por parte del FONCA, la UNAM, TELMEX, MIT, Universidad de Washington, DXARTS, IMEB-Bourges, Centro Multimedia, Transitio_MX y Leonardo, entre otros. En 2013 fue distinguido como miembro de Sistema Nacional de Creadores de Arte. Durante ese periodo realizó el proyecto Sonoridad Delta y desde entonces trabaja temas relacionados con el sonido urbano.

Es profesor titular de tiempo completo de la carrera de Arte y Comunicación Digitales de la Universidad Autónoma Metropolitana Unidad Lerma. De 2013 a 2015 fue profesor de tiempo completo del Tecnológico de Monterrey. Durante los últimos años se ha dedicado a la docencia en el campo de la creatividad digital. Su libro, *El proceso en la producción de arte digital* está en proceso de publicación.

En 2012 y 2013 fue director del departamento de tecnología, medios e interacción del Centro de Cultura Digital de Conaculta. Realizó un doctorado en Arte Digital y Medios Experimentales (DXARTS) en la Universidad de Washington bajo la dirección de Juan Pampin

y Richard Karpen. Su tesis final consistió en la grabación y manipulación sonora de sonidos de ventilas hidrotermales y la creación de *Axial*, una instalación sonora a partir de dichos sonidos. Durante este periodo desarrolló el software para creación multimedia Juum, realizó conciertos de improvisación electroacústicos solistas y con el Duo Juum, y produjo varias obras multimedia, entre ellas las esculturas sonoras con pianos acústicos *Piano delator* y *Metáforas para pianos muertos* así como trabajos interactivos como *Swarms*, *Exhalation*, *Water* y *Ojos te vean*. Estos trabajos se han presentado en diversas galerías, exhibiciones y festivales, entre ellos Jack Straw Sound Gallery, MacLeod Residence, Transitio_MX, Sonorities, FONCA XX años y Kirkland Art Center.

Entre 2004 y 2006 realizó un DEA en Ciencias de la Computación y Comunicación Digital en la Universidad Pompeu Fabra en donde fue miembro del Grupo de Tecnología Musical. En la UPF fundó el ensAmble Crumble y realizó análisis computacional de su música. Al mismo tiempo colaboró con el grupo de Sistemas Interactivos ayudando en la etapa inicial del reacTable.

Entre 2002 y 2004 realizó la maestría en Artes y Ciencias en el Media Laboratory del Massachusetts Institute of Technology en el grupo Ópera del Futuro en donde desarrolló el sistema IMPI para analizar “gestos” gráficos y convertirlos –en tiempo real– en notación musical tradicional. El sistema es una herramienta para improvisadores y permite coordinar grupos de cámara para realizar improvisaciones colectivas. El sistema fue estrenado durante el Experimental Musical Instrument Workshop dirigido por John Zorn. Algunos de sus maestros en el MIT fueron Tod Machover, Barry Vercoe, John Maeda, Christopher Csikszentmihályi y Joe Paradiso. Colaboró en el proyecto pedagógico *Toy Symphony* presentándose en Boston, Nueva York y Lisboa.

Realizó sus estudios de licenciatura en piano en la Escuela Nacional de Música de la UNAM. Algunos de sus maestros fueron Alejandro Escuer, Eva del Carmen Medina, Andrés Acosta, Tere Frenk, Salvador Rodríguez y Mauricio Nader. Estudió composición con el Dr. Julio Estrada dentro del Laboratorio de Creación Musical de la ENM, y música electrónica con Nick Didkovsky en la Universidad de Nueva York. Durante este periodo tocó como solista, como acompañante y en ensambles de cámara en México y en el extranjero y participó en varios festivales y congresos entre ellos el Foro Internacional de Música Nueva y el congreso

Internacional de Música Electroacústica de Brasil. En 2000 fundó el Trío de improvisación de Música Electroacústica NICROM con el que realizó una temporada de conciertos en la Ciudad de México. Para más información favor de visitar <http://hugosolis.net>. ■ ■ ■

Referencias

- “Carrera, música y tecnología”, *La Jornada*. <http://www.jornada.unam.mx/2016/03/22/sociedad/034n1soc>, 03 2016.
- Abad, Antoni (2006), Barcelona*accessible. <http://megafone.net/barcelona>.
- Alberro, A. y B. Stimson (1999), *Conceptual Art: A Critical Anthology*. MIT Press. MIT Press.
- ATMEL (2009a), *Mega 168 Datasheet*. ATMEL.
- ATMEL (2009b), *SerialNet User Guide*. ATMEL.
- Baker, Stephen (2009), *The Numerati*. First Mariner.
- Banzi, Massimo (2009), *Getting Started with Arduino*. Make O’Reilly Media.
- Benítez, Mónica (ed.) (2015), *Comunidades y contextos en las teorías y prácticas artísticas contemporáneas*. Juan Pablos.
- Benítez, Mónica, et. al. (2014), *Comunidades alternas; espacio, memoria y archivo en el arte relacional*. Universidad Autónoma Metropolitana.
- Christiane Paul (2008), *Digital Art, World of art*. Thames and Hudson, 2a edición.
- Colson, Richard (2007), *The Fundamentals of Digital Art*. AVA.
- Crone, Timothy J. et al (2006), “The sound generated by mid-ocean ridge black smoker hydrothermal vents”. *PLoS ONE* 1.
- Didkovsky Nick y Philip L Burk (2001), “Java music specification language, an introduction and overview”. *ICMC*.
- Dierk et al. (2009), *Groovy in Action*. Manning Publications, 2a edición.

- Fiebrink, Rebecca (2011), *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*, tesis de doctorado, Princeton University, Princeton, NJ, EUA, enero.
- Hultén, Pontus y Frank Konigsberg (ed.) (1996), *9 evenings: theatre and engineering. Experiments in Art and Technology*. The Foundation for Contemporary Performance Arts.
- Jehan, Tristan (2005), *Creating Music by Listening*. Tesis de doctorado, Massachusetts Institute of Technology.
- Jorda, Sergi (2005), *Digital Lutherie: Crafting musical computers for new musics' performance and improvisation*. Tesis de doctorado, Universitat Pompeu Fabra.
- Kadenze (2017), <https://www.kadenze.com/courses/machine-learning-for-musicians-and-artists>.
- Kelly, Caleb (2009), *Cracked Media; the sound of malfunction*. MIT Press.
- Khan, Osman (2017), <http://www.osmankhan.com/König>.
- Kwastek, Katja (2013), *Aesthetics of Interaction in Digital Art*. MIT Press.
- Lev, Manovich (2001), *The language of new media*. Leonardo Book. MIT Press.
- Levin, Golan (2010), *Dialtones*. <http://www.flong.com/storage/experience/telesymphony/index.html>.
- Licht, Alan (2007), *Sound Art; Beyond Music, Between Categories*. Rizzoli International Publications. Inc.
- Lieberman, Zach (2016), *Openframeworks*. <http://openframeworks.cc/ofBook/chapters/foreword.html>.
- Maeda, John (2001), *Design by Numbers*. MIT Press.
- MIT. *History of act* (2016), <http://act.mit.edu/about-act/history/>, abril.
- Morozov, Dmitry (2015), <http://archive.aec.at/submission/2015/DM/49157/>.
- Pampin, Juan (2016), *Sanctum*, abril.
- Proskurowski, G. et al. (2011), "Ocean Observing at Axial Seamount: Details of the Current OOI-RSN Design Within the Context of the April 2011 Eruption", *AGU Fall Meeting Abstracts*, diciembre.
- Reas, Casey y Ben Fry (2014), *Processing: A Programming Handbook for Visual Designers and Artists*. MIT Press, 2a edición.

- Reblitz, A. A. (1993), *Piano Servicing, Tuning, and Rebuilding for the Professional, the Student, and the Hobbyist*. Vestal Press.
- Russell Stuart y Peter Norvig (1995), *Artificial Intelligence: A Modern Approach*. Prentice Hall, Inc., Upper Saddle River, New Jersey.
- Shanken, Edward A. (2013), *Inventar el futuro: Arte - Electricidad - Nuevos Medios*. Fiction Department, Brooklyn.
- Shanken, Edward A. (ed.) (2009), *Art and Electronic Media*. Phaidon.
- Smith, Steven W. (1999), *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 2a edición.
- Solís, Hugo (2004), *Improvisatory music and painting interface*, tesis de maestría, Massachusetts Institute of Technology.
- Solís, Hugo (2006), *Understanding collective gestural improvisations; a computational approach*, tesis de maestría, Universidad Pompeu Fabra.
- Sudweeks, F., M.L. McLaughlin y S. Rafaeli (1998), *Network and Netplay: Virtual Groups on the Internet*, AAAI Press Series. AAAI Press.
- Sutton, G. (2015), *The Experience Machine: Stan VanDerBeek's Movie-Drome and Expanded Cinema*. Leonardo (Series) (Cambridge, Mass.). MIT Press.
- Tribe, Marky Reene Jana (2006), *Arte y nuevas Tecnologías*, Taschen.
- Urrutia, Jaime *et al.* (2015), *Pseudomatismos*. Exhibiton Catalog, MUAC.
- Vattimo, G. (2011), *Adiós a la verdad*. CLA-DE-MA / Filosofía. GEDISA.
- Wands, Bruce (2006), *Art of the Digital Age*. Thames and Hudson.
- Wilson, Scott, David Cottle y Nick Collins (ed) (2011), *The SuperCollider Book*. MIT Press.
- Wilson, Stephen (2002), *Information Arts; intersections of art, science, and technology*. MIT Press.
- Winkler, Todd (1998), *Composing Interactive Music: Techniques and Ideas Using Max*. MIT Press.
- ZigBe Organization (2008), "Zigbee specification document 05347r17". *Technical report*, ZigBee Standards Organization, enero.

