Introducción: programación en ChucK para artistas

Este capítulo cubre:

- ¿Por qué los artistas necesitan programar?
- ¿Qué es ChucK?
- ¿Por qué habría que programar en ChucK?
- ¿Por qué nosotros y más gente usan ChucK?
- Algunas de nuestras obras de arte con computadores

Por muchos años, las palabras músico y artista han estado cambiando su significado, rápidamente, de forma diaria, en gran medida debido a la introducción de la tecnología de los computadores. Los artistas hacen presentaciones en vivo con computadores todo el tiempo. Los que interactuan directamente con computadores como parte de sus presentaciones pueden llamarse a sí mismos DJs, artistas de computador, programadores en vivo, y una multitud de otros nombres. Muchos de estos músicos no programan o escriben software, pero un número creciente quiere tener un mayor control directo sobre su proceso y los resultados. Aprender a programar es una forma de obtener ese nivel extra de control.

Otros artistas quieren hacer nuevos instrumentos o controladores o configurar los controladores existentes como drum pads, estaciones de control de DJ, y otros por el estilo para usarlos de maneras novedosas en sus presentaciones en vivo. Otros todavía quieren producir canciones y álbumes (archivos wav y/o .mp3) como resultado final pero quieren tener un mayor control del proceso que el ofrecido por software tradicional. A

otros les gusta (o quieren) programra como la base de su proceso creativo y flujo de trabajo.

Artistas no-musicales también usan computadores al momento de crear arte, como diseñadores gráficos, animadores, editores de video, diseñadores de escenarios, escultores, y otros que usan gráficas computacionales y software de diseño. Muchos de estos usuarios de herramientas de software también quieren un mayor control sobre sus creaciones que el provisto por paquetes de software comerciales.

De forma creciente, muchos artistas multimedia crean instalaciones de arte, arte público, esculturas sonoras, o paisajes sonoros. Estos nuevos trabajos usan combinaciones de sonido, sensores, gráficas, video y pantallas para crear ambientes interactivos. Estas piezas pueden hacer a la audiencia parte de la performance o experiencia. En algunos casos, miembros del público testigos de estas piezas no saben exactamente qué es lo que los artistas/performers están haciendo o controlando, pero el descubrimiento, aprendizaje e interacción son muchos veces el punto y la recompensa. Aunque este libro no te enseña específicamente cómo hacer todo tipo de obra de arte, muchas de estas formas de nuevos medios involucran sonido computarizado y música, y aprender herramientas específicas para la creación y manipulación de sonido en el computador es precisamente sobre lo que trata este libro.

Mucha gente llama a estos tipos nuevos de arte y oportunidades de performance como sistemas de arte mediados por computador. En todos los casos, el músico o artistanecesita saber cómo programar o colaborar con alguien que puede. Aprenderás a programar a lo largo de estos capíutlos, ejemplos y ejercicios de este libro.

Este movimiento de popularidad en aumento de usar computadores como nuevo medio para fines creativos motivó a nuestro equipo a escribir este libro para hacer más fácil y más entretenido para todos el aprender a escribir sus propios programas. Específicamente, queremos enseñarte a programar a través de código que se traduce en música y sonido, y lo logramos a través de un lenguaje de programación llamado ChucK, que está especialmente diseñado para sonido y música.

Empezamos con una discusión en por qué creemos que músicos y artistas necesitan aprender a programar. Explicamos por qué creemos que ChucK es un gran lenguaje para empezar a aprender. Para finalizar describimos como la programación le ha permitido a los autores y a otros a crear nuevos trabajos usando ChucK y sus lenguajes antecesores.

0.1 ¿Por qué músicos y artistas necesitan programar?

Como lo mencionamos anteriormente, muchos artistas están contentos con sistemas de software tradicionales y controladores para presentaciones en vivo tradicionales. Y hay muchos que solo usan computadores para producir productos estáticos finales en la forma de archivos .wav/.mp3, CDs o colecciones de canciones, bandas sonoras para videos, y más. Un gran número de artistas están contentos con aprender y usar los paquetes y herramientas de fuentes comerciales o gratuitas.

Pero hay muchos, y estamos apostando a que eres uno de ellos, que quieren más. Quizás estás llegando a este libro con una gran idea (o muchas grandes ideas) y quieres las herramientas que te ayuden a

hacerla(s) realidad. Quizás estás buscando cambiar de dirección en tu práctica artística. O quizás ya sabes cómo programar en un lenguaje como Java, pero encuentras que no puedes hacer lo que quieres hacer.

Algunos creen que aprender a programar les ayudará a encontrar trabajo. Aunque no podemos prometer que encuentres trabajo apenas termines este libro, podemos decirte que tenemos buenos trabajos, y nuestros estudiantes tienen buenos trabajos (al menos los que los quieren), debido en no menor parte a su habilidad para programar y resolver problemas con computadores.

Nosotros, en nuestros proyectos individuales, obras de arte, y clases, hemos usado ChucK y otros lenguajes de música/arte con computadores durante años. Algunos de nosotros hemos programado, y todavía lo hacemos, en múltiples lenguajes. Seas o no ya un programador, tu pensamiento cambiará tras trabajar con los ejemplos y ejercicios de este libro. Pocos pintores han sufrido tras aprender más sobre la química y la física detrás de sus pinturas, lienzos, brochas y solventes. Incluso artistas autodidactas han desarrollado y usado conocimientos naturales de los procesos tras su práctica artística. Aprender programación es similar para el artista digital. Cualquier artista que sabe uno o más lenguajes de programación, incluso uno que no escribe código a diario, desarrolla una mejor intuición de lo que está pasando cuando navegan un menú, seleccionan un item o ven cómo la barra de progreso se mueve.

Estamos seguros de que tras finalizar estse libro serás capaz de lograr muchas cosas que quieres hacer y probablemente muchas más de las que creíste posibles. El poder de ChucK te sugerirá nuevas cosas que hacer, y ahora sabrás cómo hacerlas - o cómo resolverlas. Incluso si usas mucho software comercial, te mostraremos nuevas maneras de controlar e

interactuar con él usando ChucK. Es como ser capaz de ponerle un nuevo motor a tu auto.

0.2 ¿Qué es Chuck? ¿Por qué es diferente?

Si estás acostumbrado a trabajar con herramientas populares de sonido en tu computador, estarás acostumbrado a conectar dispositivos de procesamiento y síntesis de sonido, administrar varias pistas e instrumentos, edición de sonido con cortar/copiar/pegar, mover perillas y faders virtuales en estaciones de trabajo de audio digital (DAW, por la sigla en inglés de Digital Audio Workstation), entgre otros. En este libro, aprenderás un lenguaje de programación, ChucK, que te permitirá esencialmente hacer cualquier tarea, pero necesitarás escribir algunas líneas de código (texto) para lograr tus objetivos. Los primeros ejemplos pueden parecer más difíciles que las herramientas con las que estás acostumbrado a hacer sonido y música, pero pronto estarás haciendo cosas que nunca creíste o supiste que fueran posibles. En este sentido puedes considerar a ChucK como una versión eléctrica de tus herramientas manuales. Pronto tus resultados te harán preguntarte cómo has vivido hasta ahora sin ellos.

ChucK es un lenguaje de programación diseñado específicamente para síntesis de sonido y creacion musical en tiempo real. Tiempo real significa que ChucK sintetiza el sonido a medida que lo vas escuchando (en vez de reproducirlo desde un archivo de audio), a menudo en respuesta a señales y gestos provenientes del mundo externo al computador. Los gestos para controlar sonido pueden incluir escribir en el teclado, mover el ratón del

computador, manipular una palanca de mando u otro controlador de juegos, o tocar las teclas de un teclado musical conectado a tu computador. Chuck es también bueno para controlar y/o interactuar con casi cualquier tipo de medios computacionales y arte en tiempo real, como gráficas, robots, o cualquier dispositivo capaz de comunicarse con tu computador.

ChucK fue diseñado específicamente para permitir y promover programación sobre la marcha, lo que significa que puedes añadir, quitar, modificar, editar y superponer segmentos de código en cualquier momento, escuchando los resultados instantáneamente, sin interrumpir otros sonidos siendo sintetizados y escuchados. Esta es una de las maneras principales en que ChucK difiere de todos los otros lenguajes, lo que lo hace extremadamente entretenido de aprender y usar, porque puedes probar cosas e inmediatamente escuchar los resultados. La mayoría de los otros lenguajes requieren compilar, ejecutar y corregir errores en una forma que no te permite escuchar inmediatamente lo que estás haciendo. La mayoría de los lenguajes de programación, como C, C++ o Java no fueron diseñados específicamente desde cero para tareas de sonido, música o en tiempo real. ChucK tiene como prioridad hacer sonido en tiempo real.

Si sabes otros lenguajes de programación como Java o C++, o incluso otros lenguajes de programación para música/sonido como Csound, SuperCollider, JSyn, Max/MSP o PD (Pure Data), pronto verás que ChucK es realmente diferente. Es más expresivo y poderoso en manipulación de tiempo y sonido que las interfaces gráficas de Max/MSP y PD, brindándote mayor acceso tras bambalinas que estos otros lenguajes y sistemas. Comparado a otros lenguajes basados en texto para música/sonido como SuperCollider o Csound, ChucK es generalmente más breve, requiriendo

menos código (línea de texto escritas) que estos otros lenguajes para lograr cualquier tarea en particular.

Si no sabes ningún lenguaje de programación, cuando hayas terminado este libro, te será más fácil aprender Java, C, C++ y cualquier otro lenguaje que desees. ChucK es diferente a otros lenguajes, por supuesto, pero comparte muchas cosas que lo hacen similar y reconocible a programadores de cualquier otro lenguaje.

Otra gran característica de ChucK is que es de código abierto (no es secreto in protegido por licencias, contraseñas, llaves, entre otros), y está disponible de forma gratuita en las plataformas computacionales más populares, incluyendo Mac OS X, Windows y Linux. Código libre significa que la comunidad de usuarios de ChucK puede tener acceso directo al proceso de construcción de un mejor ChucK a futuro. También significa que la obtención y uso de ChucK no implica ningún costo.

0.3 ¿Por qué programar en ChucK?

Una herramienta, que es una manera de concebir un lenguaje de programación, no puede dejar de formar la mentalidad del usuario, y naturalmente sugiere maneras de lograr ciertas tareas. Y como cualquier herramienta, un lenguaje de programación debería cambiar la manera en que piensas y haces tareas. ChucK definitivamente presentea una manera disinta de programar sonido y música. Aunque existe un equilibrio entre hacer ciertas cosas simples y otras más difíciles, nuestra intención es que para el programador, las decisiones de diseño del lenguaje ayuden más que dificulten cualquier tarea.

ChucK fue creado por Ge Wang, ex estudiante de postgrado en Princeton

University bajo la tutela del coautor Perry Cook (y ahora profesor en Stanford University y coautor de este libro), para proveer una manera distinta de pensar sobre la programación de sonido y para una mentalidad de prototipado rápido (permitiendo probar muchas ideas rápidamente). Más de una década después, la investigación y el desarollo de ChucK se han intensificado La detallada crónica de Ge de la historia, motivación y la gente detrás de ChucK puede ser encontrada en el prefacio de este libro. Para mñas detalles de por qué deberías aprender y usar ChucK, aquí van unas razones adicionales:

- Todo gira en torno al tiempo. El tiempo está en el centro de cómo ChucK trabaja y cómo trabajas con ChucK para hacer sonido. Como programador, tú especificas cómo te mueves a través del tiempo y tomas control en momentos específicos, y el sonido esencialmente ocurre - convenientemente, durante la cantidad de tiempo precisa que te has movido. ¿Por qué tanto énfasis en el tiempo? El sonido es un fenómeno basado en el tiempo; sin flujo de tiempo, no habría sonido. Controlando cómo y cuándo haces cosas a través del tiempo, tienes una manera distinta y poderoso de trabajar con sonido en cada nivel - cada molécula de él.
- Es texto, plano y simple. Aunque la programación con texto puede inicialmnete parecer más abstracta o compleja que una representación gráfica, es posiblemente más fácil una vez que empiezas a añadir matices expresivos y lógica a tu código (lo que invariablemente tendrás que hacer). Poco es escondido o inferido. Las partes importantes están a simple vista: por ejemplo, cómo el tiempo fluye en el programa. Al mismo tiempo, muchos aspectos mundanos son atendidos tras bambalinas: temporización, entrada y salida de sonido en tiempo real, organización de todos los generadores de sonido,

entre otros. La legibilidad es una meta central de diseño del lenguaje, y eso lo hace también una buena herramienta de aprendizaje.
- Es entretenido e inmediato. ChucK fue diseñado para ser un lenguaje y un entorno divertido para trabajar y experimentar. Puedes sintetizar sonidos, hacer automatizaciones fantásticas, mapear gestos físicos (por ejemplo, con controladores) a sonidos, conectar distintos computadores e incluso usar análisis de señales para (computacionalmente) encontrarle un sentido al sonido.

0.4 Arte con computadores con ChucK y pre-ChucK

La meta de esta sección es presentar ejemplos de obras de arte que median con computadres que han sido creados usando programación - creación de instrumentos msuicales, controladores, conjuntos y sistemas. Una manera tradicional de construir nuevos instrumentos es observar instrumentos existentes y encontrar maneras de aumentarlos o mejorarlos. Muchas de nuestras experiencias pertenecen a esta área. Desde la Cook/Morrill MIDI Trumpet del año 1988 (no utiliza ChucK, porque todavía no existía, pero sí usaba el predecesor de ChucK, STK, que ahora está incluido en ChucK) hasta la eSitar de Kapur del año 2004, pasando por las Laptop Orchestras del año 2005 hasta el presente, y los instrumentos musicales basados en dispositivos móviles y más allá, nosotros y nuestros locos amigos hemos creado una variaedad de sistemas mediados por computador basados e inspirados en instrumentos y grupos musicales tradicionales. En algunos casos hemos añadido sensores al instrumento, respetando sus capacidades inherentes de producción sonora. En otros casos, hemos destripado las partes acústicas del instrumento y las hemos

llenado con procesadores, sensores y parlantes, dejando solo la coraza y forma para sugerir la interacción (experiencia). En esta categoría se incluyen el DigitalDoo, SqueezeVox (acordiones, Perry Cook y Colby Leider), la ETabla y el EDholak (Ajay Kapur), BoSSA (Bowed Sensor Speaker Array, por Dan Trueman y Perry Cook) y el SBASS (Sensor Bass, por Curtis Bahn). La figura 0.1 muestra algunos de estos instrumentos aumentados por computador. en todos estos casos, necesitamos escribir código para que un microchip o un teléfono celular transformara un gesto en una experiencia musical.

La proliferación de dispositivos móviles poderosos como el iPhone y el iPad promovieron la reimaginación de una era móvil de instrumentos musicales tradicionles, como la Ocarina para iPhone (Ge Wang / Smule), así como también nuevas interacciones musicales móviles. Con millones de usuarios a lo largo del mundo, Ocarina y otras aplicaciones móviles han usado ChucK para síntesis.

Otra manera de hacer sistemas de performance musical completamente nuevos es poner sensores en bailarines u otros artistas que tradicionalmente no ejecutan música. La figura 0.2 muestra PikaPika, un personaje bailarín inspirado en Anime creado por Tomie Hahn. PikaPika está lleno de sensores, transmisores inalámbricos, amplificadores y parlantes. El hardware de Pika llamado SSPeaPer (Sensor-Speaker Performer) fue creado y programado por Curtis Bahn. La figura 0.2 también muestra a Raakhi Kapur, un bailarín indio, vistiendo sensores en sus muñecas que controlan robots musicales en una producción de la KarmetiK Machine Orchestra, protagonizada por robots musicales y músicos humanos (más información pronto). Mucha programación fue necesaria para lograr mapear los gestos de estos bailarines a respuestas sonoras. Pika usa STK (incluido en Max/MSP), y el sisteam de Raakhi usa

ChucK para síntesis y control de robots.

Otro medio para producir piezas de arte y música interesantes es poner sensores en objetos que no son tradicionalmente asociados a la performance musical. La figura 0.3 muestra utensilios de cocina musicales; el JavaMug de Perry Cook y el Filliup Glass encima de la mesa musical de P-Ray's Café. También se muestra un prototipo temprano de un zapato de tap musical mediado por computador llamado TapShoe. Convertir estos objetos de uso cotidiano en instrumentos musicales obviamente requiere programación, y una vez más, mapeo, para hacerlos sónicamente interesantes. Todos estos dispositivos emiten señales que el computador usa para producir sonido o controlar otros objetos/procesos, y ChucK es el lugar más fácil donde hacer este mapeo. En uno de nuestros programas favoritos de ChucK, Perry convirtió el JavaMug (taza de café computarizada mostrada en la figura 0.3) en una trompeta digital.

Las posibilidades de conjuntos completos de computadores y/o otros grupos de gran escala mediados por computador conformaron la Princeton Laptop Orchestra (PLOrk), seguida de la Stanford Laptop Orchestra (SLOrk), y el conjunto compuesto de humanos + robots llamado KarmetiK Machine Orchestra. Todos estos conjuntos confían fuertemente en performers humanos aumentados con sensores, redes y ChucK. Además, cada instrumento de la orquesta presenta sonido local al intérprete a través de parlantes individuales hemisféricos. Esto signfica que no existe un ruteo de los computadores a un mezclador de audio y al sistema de sonido del recinto, sino que cada computador se conecta a un parlante multicanal individual (seis parlantes en un arreglo hemisférico) ubicado al lado de cada intérprete. Es así que el sonido del conjunto es la mezcla acústica, en la pieza o sala de conciertos, de los sonidos individuales de cada intérprete, tal como en el caso de una orquesta tradicional. La figura 0.4

muestra la orquesta original de PLOrk (con estudiantes universitarios de primer año de pregrado), la primera orquesta de teléfonos móviles en Stanford, SLOrk, y la KarmetiK Machine Orchestra.

Estos ejemplos solo rasguñan la superficie de los objetos artísticos posibles cuando sabes cómo hackear. Aquí usamos hack en el mejor sentido de la palabra, combinando el sueño de los inventores de tener la habilidad de modificar, construir, experimentar y posiblemente lo más importante, programar.

0.5 Resumen

En este capítulo, esperamos haberte convencido de que aprender a programar es importante para los artistas digitales y que ChucK es una muy buena manera de aprender a programar. Gastamos un poco de tiempo hablando sobre lo que ChucK es, un poco en cómo fue diseñado, y por qué nosotros y muchos otros creemos que es el mejor lenguaje de programación para hacer arte digital. Por ahora, las cosas a recordar son las siguientes:

- Los artistas, especialmente los artistas digitales, pueden obtener beneficios al aprender a programar. Puedes construir tus propias herramientas y creaciones totalmente nuevas, y la programación te da ideas que no habrías tenido sin haber tenido habilidades en algún lenguaje de programación.
- Chuck, como lenguaje, es diferente de otros lenguajes y programas de música. Fue diseñado desde cero para soportar poderosamente el control sobre el tiempo y el sonido, dispositivos de entrada al mundo exterior (palancas de mando, trackpads, entre otros) y de redes de

computadores.
- Para aquellos que ya saben un lenguaje de programación, creemos que encontrarás que ChucK te permite hacer muchas cosas que antes creías difíciles o imposibles en otros lenguajes, y que puedes hacerlas de forma rápida.

Te hemos mostrado algunos de los tipos de proyectos que hemos creado usando computadores y programación, pero estos solo rasguñan la superficie de lo que puede ser hecho con una herramienta poderosa como ChucK.

¡Esperamos que estés convencido, inspirado y listo para empezar a aprender a programar con ChucK!