

SÍNTESIS DE SONIDO EN DISPOSITIVOS MÓVILES

Antonio Zimmerman¹, Claudio Aciti^{1,2}, Pablo Cosso¹, Esteban Sebastiani¹, Fabricio Chungo¹, Juan Amusategui¹, Francisco García¹, Catriel Nievas¹, Romina Gomez¹, Alan Achaval¹, Germán Torós¹ y Diego Leguizamón¹

¹ Departamento de Ciencia y Tecnología
Universidad Nacional de Tres de Febrero (UNTREF)
Valentín Gómez 4752 (+54-011-47593528) - Caseros - Buenos Aires

² Instituto INTIA - Facultad de Ciencias Exactas
Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA)
Pinto 399 (+54-249-4439650) - Tandil - Buenos Aires
caciti@exa.unicen.edu.ar

Resumen

Este proyecto se propone investigar temas relacionados a síntesis de sonido en dispositivos móviles. Las aplicaciones de sintetizadores digitales permiten emular el sonido analógico. Aunque es necesario ahondar en temas como la velocidad de reproducción, el jitter, la latencia "touch to sound", el timbre, entre otros. El equipo de trabajo, con base en UNTREF, está integrado por investigadores y alumnos de las especiales Computación, Música, Sonido y Artes Electrónicas. Hasta el momento, se cuenta con un desarrollo principal de un sintetizador, y varias versiones que permiten comparar entre diferentes arquitecturas de hardware, modelos de dispositivos móviles y lenguajes de programación.

Palabras claves: procesamiento digital de sonido - dispositivos móviles - sintetizador modular analógico

Contexto

Un sintetizador analógico modular es un sistema electrónico desarrollado para la creación y composición del sonido mediante

la manipulación de sus parámetros básicos. Utiliza circuitos y señales analógicas para su generación. De esta forma el timbre de un sonido es modelado utilizando diferentes tipos de subsistemas, conformados en tres grandes grupos: Generación, Procesamiento y Sistemas de Control [1,2]. El primer sintetizador fue creado por Robert Moog en el año 1965 y este era capaz de utilizar módulos separados para la generación de una amplia variedad de sonidos mediante técnicas de síntesis como la aditiva, sustractiva y FM, entre otras [3]. Un sintetizador modular es una combinación de módulos separados que el usuario interconecta pero que son en conjunto controlados por tensión o voltaje. Un sistema modular puede variar en tamaño y complejidad en relación a la cantidad de módulos disponibles, y las necesidades del usuario. Los sistemas modulares le brindan al usuario total libertad para realizar las conexiones entre las partes que lo componen. Esta configuración de conexiones es llamada "patch".

Los sintetizadores digitales, a diferencia de los analógicos, generan y controlan sus

sonidos por medio de señales digitales. Utilizan técnicas de procesamiento digital de señales para su creación [4]. La señal es procesada por algoritmos y finalmente se la convierte en una señal analógica. Algunos ejemplos de sintetizadores digitales son el Synclavier, Yamaha, Roland líneas, etc.

Los avances en la tecnología, lenguajes de programación y técnicas de procesamiento digital de señales hacen posible el desarrollo de un sintetizador modular digital para dispositivos móviles [5][6]. Para ello se utilizan lenguajes de programación de alto nivel como Pure Data [7][8][9], Max-MSP [10], Faust [11], Super Collider [12], entre otros.

En la actualidad, los dispositivos móviles han cambiado la forma en que las personas se relacionan entre sí, ya que son un producto clave de la tecnología de la información [13][14]. Con relación a la música, permiten desarrollar nuevas posibilidades que facilitan al momento de experimentar nuevas formas de expresión de un modo mucho más accesible. Con los sensores adecuados se puede generar nuevas formas de interpretar música, permitiendo que un dispositivo de uso masivo pueda ser utilizado por un público más amplio.

Introducción

En un sintetizador, el usuario puede elegir el tipo de onda de generación del sonido. Los tipos de onda más utilizados en síntesis son: sinusoidal, cuadrada, triangular y diente de sierra. En el dominio digital no pueden existir señales continuas ya que requerirían infinita cantidad de información. Representar una señal continua implica convertirla a una forma discreta; para ello es necesario realizar un muestreo de la información. Según el Teorema de Nyquist, para no perder información en la conversión, la frecuencia de

muestreo debe ser igual o mayor que el doble de la frecuencia máxima [15][16][17].

En un sintetizador modular, el usuario determina la configuración de conexiones con un criterio artístico y no meramente por limitaciones técnicas. Esto posibilita y fomenta la libertad de exploración de diversas posibilidades sonoras dentro de los distintos métodos de síntesis [18][19][20].

El objetivo principal de este proyecto es el estudio, diseño y desarrollo de aplicaciones que sintetizen sonido en dispositivos móviles que emulen fielmente un sintetizador modular analógico. Es necesario realizar estudios y comparativas de velocidad, jitter, timbre y latencia "touch to sound", entre diferentes dispositivos, sistemas operativos y lenguajes de programación.

Metodología de trabajo

Se utilizan metodologías ágiles poniendo énfasis en el desarrollo iterativo e incremental.

Captura de requerimientos: Se realiza un relevamiento de sintetizadores analógicos; sus distintos componentes, interacciones y comportamientos. Se releva la resolución del manejo de interfaz de usuario en sintetizadores digitales disponibles para dispositivos móviles.

Sesiones de Brainstorming del equipo interdisciplinario para definir el alcance y las metas de cada etapa.

Líneas de Investigación, Desarrollo e Innovación

Este proyecto se desarrolla en la UNTREF, con un equipo de trabajo multidisciplinario, formado por alumnos y docentes provenientes de las carreras Ingeniería de Sonido, Licenciatura en Artes Electrónicas,

Licenciatura en Música e Ingeniería en Computación.

Hasta el momento se diseñó y se cuenta con un desarrollo avanzado de una aplicación para dispositivos móviles que busca emular fielmente un sintetizador modular analógico. Dicha aplicación funciona en plataformas Android, y fue escrito en lenguaje Java y Pure Data.

Entre las líneas de investigación se destacan:

- La implementación de diferentes aplicaciones que requieren de un estudio previo, puesta en funcionamiento y comparación de diferentes lenguajes de programación. Y las comparativas con apps actuales, de licencia libre o privativa.
- El estudio de jitter y de la latencia "touch to sound" de las aplicaciones entre diferentes lenguajes de programación, diferentes marcas y modelos de dispositivos móviles y diferentes versiones de sistemas operativos.
- La creación y estudio de funcionamiento de sonidos preconfigurados en diferentes plataformas y dispositivos móviles.
- El diseño de interfaz de usuario que permita a los usuarios utilizar aplicaciones de este estilo de forma intuitiva, fácil de comprender, que se adapte a diferentes tamaños de pantallas, y que sea capaz de responder a velocidades razonables independientemente del dispositivo donde esté corriendo.
- El diseño y desarrollo de los módulos generadores de audio, de tal manera que puedan combinarse entre sí y funcionen en diferentes dispositivos móviles.

Resultados

Como se dijo anteriormente, se cuenta con un diseño y desarrollo avanzado de un sintetizador modular llamado Synth3F¹, que corre en Android y está programado en Java y Puredata.

Se comparó el nivel de aliasing de los generadores de ondas de las dos APIs utilizadas para sintetizar audio en Android. La primera de ellas es la basada en libPd, que permite embeber un código o patch escrito en el lenguaje Pure Data. La segunda es la API provista por el lenguaje Faust, que compila el código en lenguajes Java y C++ para ser implementados dentro de la aplicación. La comparación entre ambos motores de audio muestra un resultado notablemente favorable para el caso de Faust. La síntesis desde Pure Data muestra componentes no deseados incluso en el caso de la onda senoidal. En general, estos componentes se encuentran distribuidos a lo largo del espectro y sus picos más altos tienen una diferencia cercana a los 20 dB respecto a la fundamental.

Se realizaron mediciones de latencia "touch to sound" en diversos dispositivos Android y se comparó su comportamiento en distintas versiones de la aplicación. Primero, se buscó obtener la medición de latencia de manera óptima. Para esto se utilizó un micrófono de medición apuntando hacia el dispositivo bajo prueba. El transductor se conectó con una placa de audio, y luego a una Digital Audio Workstation (DAW) para su análisis posterior. De esta forma, el transductor logra captar dos impulsos diferentes: el primero referido al golpe del dedo sobre la pantalla del dispositivo, y el segundo relacionado con el sonido emitido por el parlante del mismo, una vez que el proceso de la señal fue realizado. Los parámetros de la aplicación en cuestión se ajustaron para que el sonido emitido posea un valor de ataque bajo (idealmente, lo más cercano posible a 0 ms). De las mediciones obtenidas de los tiempos de latencia "touch to

¹ <https://github.com/UntrefSintetizador>

sound" obtenidas de la aplicación propia contra aplicaciones comerciales, se puede observar que efectivamente los valores son similares. Es notable destacar esta comparación, ya que coloca a la aplicación (en referencia a tiempos de latencia "touch to sound") al nivel de aplicaciones comerciales actuales.

Además, se realizó la compilación de archivos de Faust en Android Studio con el fin de utilizar módulos de Faust en lugar de los de Pure Data ya que el mismo cuenta con antialiasing y una mejor calidad de audio.

Formación de Recursos Humanos

El equipo de trabajo es dirigido por un profesor con maestría de la carrera Ingeniería en Computación, quien se especializa en procesamiento de señales y espera terminar su doctorado en Matemática Computacional e Industrial de la UNCPBA en 2020. La codirección está a cargo de un Licenciado en Música, quien se especializa en aplicaciones para dispositivos móviles. El resto del equipo de docentes está compuesto por un Lic. en Análisis de Sistemas, un Lic. en Artes Electrónicas, un Lic. en Música y un Ing. de Sonido.

También lo integran 8 alumnos avanzados de la UNTREF, dos de la carrera Ing. en Computación, dos de Lic. en Artes Electrónicas, dos de Ing. de Sonido y dos de Lic. en Música. Tres de ellos están realizando su trabajo final de grado en temas relacionados y se espera que continúen con sus posgrados en la temática.

Referencias

[1] Austin, Kevin. *A Generalized Introduction to Modular Analogue Synthesis Concepts*. Montréal, Canada. 2016
 [2] Rudi Esch. *Electri_City: The Düsseldorf School of Electronic Music*. Omnibus Press. Germany. 2016
 [3] Russ, M. *Sound Synthesis and Sampling*.

3ed. Focal Press. 2009
 [4] J G Proakis, D G Manolakis. *Digital signal processing - Principles, algorithms and applications*. Pearson. 2007
 [5] Miller Puckette. *Pure Data: Another Integrated Computer Music Environment*. Proceedings of the Second Intercollege Computer Music Concerts, 37-41, 1996.
 [6] Brinkman, Peter. *Making Musical Apps*. Sebastopol, CA: O'Reilly Media, Inc. 2012.
 [7] Pure Data (Pd). Pd Community Site. [online] Disponible en: <https://puredata.info/>.
 [8] P. Brinkmann, P. Kirn, R. Lawler, C. McCormick, M. Roth, and H.-C. Steiner. *Embedding Pure Data with libpd*. Pure Data Convention Weimar. 2011.
 [9] LibPD [online] Disponible en: <http://libpd.cc/about/>
 [10] Sitio web de MAX-MSP <https://cycling74.com/> accedido en Marzo de 2020.
 [11] Sitio web de FAUST. <https://faust.grame.fr> accedido en Marzo de 2020.
 [12] Sitio web de Supercollider. <https://supercollider.github.io/> accedido en Marzo de 2020.
 [13] The Economist [online] Disponible en: <https://www.economist.com/news/briefing/21645131-smartphone-defining-technology-age-truly-personal-computer>
 [14] GartnerInc.: Technology Research [online] Disponible en: <http://www.gartner.com/newsroom/id/360987>
 [15] Alan Oppenheim. *Señales y Sistemas*. Prentice Hall. 1998.
 [16] Lyons. *Understanding Digital Signal Processing*. Prentice Hall. 2001
 [17] Oppenheim, Schafer. *Discrete time signal processing*. Prentice Hall. 2009
 [18] Alan Bloom. *The ARRL Handbook for Radio Communications*, Chapter 8: Modulation. American Radio Relay League. 2010
 [19] Simon Haykin. *Communication Systems*. 4th Ed. John Wiley & Sons. New York. 2001
 [20] Puckette, Miller. *Theory and Techniques of Electronic Music*. WSP. 2007