

Interconexión y código

Introducción a la música por computadora en red

Hernani Villaseñor Ramírez, 2014

Centro Multimedia, CENART, México D.F.

La música por computadora en red puede definirse como una práctica donde existe una interconexión entre las computadoras de los miembros de un ensamble, banda u orquesta de *laptops*. Una práctica musical donde la comunicación, la composición, la improvisación con algoritmos y el trabajo en grupo tienen varios niveles de interacción.

La interconexión en red establece la posibilidad de comunicación mediante protocolos informáticos y de comunicación. Actualmente el protocolo mas usado en la música para este fin es OSC¹. La comunicación entre computadoras se establece al conectarnos a una red y al conocer la dirección IP propia y la de las demás personas conectadas a la red, una de las estrategias usadas en esta práctica es la de *live coding* colaborativo.

Historia

En el Área de la Bahía de San Francisco alrededor de los 70s, *The League of Automatic Composers* y *The Hub* fueron dos grupos de compositores pioneros en trabajar e investigar la música mediante el uso de computadoras conectadas en red con fines de composición y ejecución de música electrónica.

Más tarde en Alemania el ensamble de computadoras PowerBooks UnPlugged desarrollaron Republic para tocar conectados en una red inalámbrica y usando solo las bocinas integradas de sus *laptops*.

Algunos ensambles y bandas de computadoras recientes son Slub, Republic 111,

¹ Open Sound Control es un protocolo de comunicación entre computadoras, sintetizadores y varios dispositivos dentro de una red.

Benoît and the Mandelbrots, BiLE, Glitch Lich y en México podemos mencionar al Colectivo Radiador y a LiveCodeNet Ensemble.

Herramientas

Para conectarnos a una red local e improvisar con código el programa SuperCollider es una buena opción, así mismo su librería JITLib² nos facilitará programar y modificar código sonoro con mucha flexibilidad de cambiar los algoritmos al vuelo, la clase *History* nos facilitará establecer una vía de conexión y visualización del código propio y de las distintas personas conectadas a la red.

Como establecer una red local y conceptos para trabajar en red

Para establecer una red de manera local contamos con dos opciones, la primera es conectar las computadoras a un *router* mediante cables ethernet, la segunda es crear una red inalámbrica en una de las computadoras o de igual manera con el uso de un *router*. En ambos casos necesitamos conocer la dirección IP que es asignada a la computadora y el puerto del programa.

Una dirección IP es la dirección que identifica nuestra computadora o dispositivo dentro de una red. El puerto del programa se refiere al punto específico donde mandaremos información.

Mensajes OSC

El protocolo OSC fue creado para establecer la conexión de distintos dispositivos multimedia en una red, con este protocolo podemos mandar mensajes de comunicación.

Los mensajes OSC en SuperCollider contienen dentro de la información que mandan una etiqueta y un valor, los valores que podemos mandar pueden ser números enteros y

² *Just in time programming library* desarrollado por Julian Rohrer.

flotantes y símbolos o palabras.

Ndef de la librería JITLib

Usaremos la clase Ndef de la librería JITLib la cual agrega a SuperCollider la posibilidad de modificar código en el momento ya que esta construida bajo la premisa de contenedores que agregan funcionalidad a los nodos antes de que estos sean ejecutados. Por ejemplo un nodo que le indicamos que suene cuando aún no ha sido programado lo que va a sonar.

Ndef o *node proxy definition* es una referencia a un contenedor o *proxy*, el cual puede ser modificado o combinado mientras esta ejecutandose. En este primer ejemplo haremos sonar una onda sinoidal a la cual le cambiaremos su primer argumento que es frecuencia.

```
Ndef(\hola).play  
Ndef(\hola, {SinOsc.ar(440, 0, 0.5)})  
Ndef(\hola, {SinOsc.ar(800, 0, 0.5)})  
Ndef(\hola).stop
```

Modelo para trabajar sonido más secuencia con Ndef con el cual podemos reprogramar el código mientras este se ejecuta.

```
// generamos sonido  
Ndef(\secuencia).play  
Ndef(\secuencia)[0]={lfreq| SinOsc.ar(frec, 0, 0.5)}  
  
// lo secuenciamos  
Ndef(\secuencia).quant_(4)[1]=\set -> Pbind(\dur, 1, \frec,  
Pseq([200,400],inf))
```

Establecer un red

Para establecer una red debemos saber algunos conceptos básicos como son la dirección IP de una computadora, el tipo de red al que estaremos conectados, si es dinámica (DHCP) o estática y si es cableada o inalámbrica.

La red cableada

En el caso de una red cableada se necesita un *router*³ y cables ethernet. Al *router* nos conectaremos mediante los cables ethernet. El *router* es el encargado de gestionar la comunicación, además genera direcciones IP al momento de conectarnos.

En el caso de una red inalámbrica el modem al que nos conectamos asigna de manera automática una dirección IP a cada dispositivo conectado. La red cableada es más segura aunque la red inalámbrica es más práctica y nos libera del uso de cables.

Mensajes OSC en SuperCollider

La música por computadora en red es un proceso de comunicación a través de mensajes utilizando una red para su distribución. Este proceso requiere del envío y recepción de mensajes. Los mensajes que usaremos para comunicarnos dentro de una red son los mensajes OSC.

En SuperCollider enviar un mensaje requiere del objeto NetAddr y para recibirlo o responder al mensaje el objeto OSCdef.

NetAddr

Esta clase se usa para enviar mensajes OSC a una dirección IP específica y a un puerto de un programa. Por ejemplo la dirección interna de tu computadora es 127.0.0.1, en

³ Es un dispositivo que sirve para conectar computadoras en red.

el ejemplo de abajo estaremos listos para enviar al puerto de SuperCollider 57120.

```
n=NetAddr("127.0.0.1", 57120)
```

Para enviar los mensajes usaremos `.sendMsg` y entre paréntesis indicamos la etiqueta y el valor. El valor puede ser una palabra o un número entero o flotante, la etiqueta sirve para dirigir el valor a un punto específico. Antes se requiere usar el objeto `OSCdef` para recibir el mensaje.

```
n.sendMsg("/etiqueta", 440)  
n.sendMsg("/etiqueta", "hola mi nombre es")
```

OSCdef

Esta clase nos permite recibir o responder a mensajes osc, esta compuesto por una llave o *key* que sirve para guardar el `OSCdef` en una colección global y se expresa mediante un símbolo, después va una función que responde a los mensajes osc entrantes y al final la etiqueta.

```
OSCdef(\miosc, {lmsg| msg[1].postln}, "/etiqueta")
```

Enviar mensajes

Para enviar mensajes declaramos un `NetAddr`, seleccionamos una etiqueta y empaquetamos un valor mediante el método `sendMsg`.

```
n=NetAddr("127.0.0.1", 57120)  
n.seneMsg("/etiqueta", 34.9)
```

La clase History

History es un objeto que establece una forma de comunicación para ver en nuestra computadora lo que otro participante esta programando, así mismo funciona como un *chat* de comunicación y es una forma de visualizar el código que los demás miembros del ensamble producen.

MandelHub

Es parte de la librería BenoitLib una extensión de SuperCollider programada por Patrick Borgeat para las necesidades de la banda de *laptop* Benoît and the Mandelbrots, tiene la particularidad de sincronizar las computadoras y la estrategia que propone es el de seguir a un líder. Benoît and the Mandelbrots programan bajo el paradigma de *live coding* basado en patrones.

Discusión

La música en red nos permite trabajar en equipo, realizar trabajos colaborativos, intercambiar ideas algorítmicas y ver procesos.

El proceso de comunicación en la música por computadora en red, si bien lo hemos definido como una forma de establecer una comunicación entre computadoras, esta es generada por los miembros del ensamble. Al momento de improvisar los gestos son importantes y nos ayudan a determinar ciertas acciones o cambios durante la presentación, la comunicación se puede establecer además mediante un *chat*. O sea, podríamos hablar de dos niveles de comunicación, el de la programación en sí, que involucra los datos compartidos entre los diferentes equipos, y la comunicación del ensamble al momento de improvisar. Estos dos niveles de comunicación no ocurren de manera desligada y uno determina al otro.

Conectarnos a la distancia

Es posible realizar conciertos a distancia mediante esta configuración ya que la única diferencia es que la comunicación se establece en una red más grande como lo es internet, esto también añade retraso en la señal y el hecho de la no presencia.

Dispositivos externos que utilizan la red

Posteriormente podemos conectarnos a una red con dispositivos que controlan el comportamiento de una programación. Así, dispositivos móviles como el teléfono celular se convierten en controladores de uso común.

Bibliografía

Bischoff, J. y Brown, Ch. (2002). *Indigenous to the net: Early Network Music Bands in the San Francisco Bay Area*.

De Campo A. y Hoelzl H. (2013). *Collaborative Network Music with the new Republic*. México DF: Centro Multimedia.

McKinney, Ch. (2013). *Networked Interfaces Workshop*. México DF: Centro Multimedia.

Rohrhuber J., de Campo A., Wieser R., van Kampen J., Hoelzl H. (2007). *Purloined Letters and Distributed Persons*. Budapest:

Rohrhuber, J. y de Campo, A. (2009). *Improvising Formalisation – Conversational programming and Live Coding*. Delatour: Ircam-Centre Pompidou.

Villaseñor, H. (2013). *Live Coding: El paradigma de la programación en vivo*. México, D.F.: Centro Multimedia.



La obra (Interconexión y código. Introducción a la música por computadora en red, de [Hernani Villaseñor Ramírez](#)), identificada por [Hernani Villaseñor Ramírez](#), está libre de restricciones de propiedad intelectual conocidas.