

CSound: software libre de síntesis sonora

Síntesis aditiva, substractiva y FM con CSound.

Durante la práctica pasada dimos nuestros primeros pasos con el CSound, un lenguaje libre de síntesis sonora. La historia de CSound se remonta a los años 1960, cuando Max Mathews desarrollaba el Music 4, un lenguaje de síntesis escrito en assembly pero que luego fue portado a Fortran. El compositor Barry Vercoe trabajó con este lenguaje y con sus sucesores y finalmente dio origen al CSound en 1985. La C de CSound proviene obviamente del lenguaje usado en el código fuente. Con el tiempo el propio Vercoe y otros continuadores ampliaron el conjunto de opcodes del CSound y añadieron la capacidad de funcionar en tiempo real.

Más información en <http://www.csounds.com/>

CSound es el motor de sonido en el muy comentado ordenador portátil OLPC . Podemos ver el software TamTam en <http://wiki.laptop.org/go/TamTam> con una interface escrita en lenguaje Python.

[One Laptop per Child <http://laptop.org/index.es.html>]

Recomendamos descargar la documentación en formato PDF si se quiere tener una referencia completa de todos los opcodes que vamos a usar en esta práctica.

<http://csound.sourceforge.net/#Documentation>

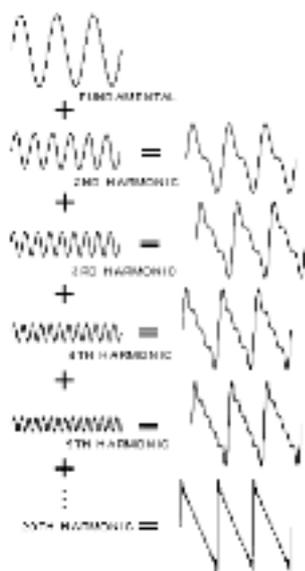
También recomendamos la relectura de la práctica 3, Introducción al CSound y repasar el significado de los parámetros de la partitura.

SEGUNDO EJEMPLO: SÍNTESIS ADITIVA

Los alumnos que hayan completado la práctica 2 habrán producido diversos sonidos usando nuestro oscilador básico. Todos ellos se habrán preguntado cómo conseguir un timbre o color sonoro más grueso o más potente o en suma más interesante.

Bueno, estamos usando la unidad básica, la onda senoidal. El resultado no se asemeja a ningún instrumento real existente.

Por las clases de teoría sabemos que casi cualquier sonido puede ser descompuesto en una suma de ondas senoidales con distintas amplitudes y frecuencias.



Ejemplo de síntesis de una onda diente de sierra.

En la práctica de introducción al editor Audacity realizamos algunos análisis espectrales de sonidos reales y anotamos los resultados en tablas. Nuestro propósito será la reconstrucción de esos sonidos usando componentes puras, ondas senoidales. Obviamente el resultado no podrá igualar al sonido de una nota de un piano de 40.000 euros ni a una nota de trompeta tocada por Wynton Marsallis [<http://wyntonmarsallis.com/>].

Pero antes debemos presentar un opcode que añadirá algo de interés a nuestros sonidos.

OPCODE **adsr**

Una de las características que distinguen a un sonido de otro es la envolvente de amplitud. En pocas palabras es la curva que "envuelve" a la onda durante el tiempo que dura el sonido.

El opcode **adsr** proporciona una manera sencilla de generar una función o curva envolvente que luego multiplicaremos por la onda a la que queremos aplicar la curva.

La sintaxis de **adsr** es :

kenv **adsr** iatt, idec, islev, irel

Ver manual PDF

Listado: "ejemplo2aditiva.csd"

```

<CsoundSynthesizer>
<CsOptions>
</CsOptions>
<CsInstruments>
sr = 44100 ;sample rate
ksmps = 1 ;control rate
nchnls = 2 ; salida de dos canales, estéreo!

;definición del instrumento 1
instr 1
iamplitude = p4
ifrequency = p5
iphase = 0

;primer parcial o fundamental
apartial1 oscils iamplitude, ifrequency, iphase
;segundo parcial, frecuencia doble, amplitud 0.8 del parcial 1
apartial2 oscils iamplitude * 0.8, ifrequency * 2 + 0.2, iphase
;añadimos pequeñas desafinaciones
apartial3 oscils iamplitude * 0.5, ifrequency * 4 - 0.3, iphase
apartial4 oscils iamplitude * 0.7, ifrequency * 8 + 0.4, iphase
apartial5 oscils iamplitude * 0.4, ifrequency * 16 - 0.5 , iphase

atotal = apartial1 + apartial2 + apartial3 + apartial4 + apartial5

iatt = 0.005 ; ataque, prueba a cambiarlo por 0.005
idec = 0.015 ; decaimiento
islev = 0.75 ; nivel de sostenimiento
irel = 0.99 ; release

;Crea una envolvente de amplitud durante la duración de la nota (p3)
kenv adsr iatt, idec, islev, irel
; multiplicamos la envolvente por la onda.
aenv = atotal * kenv
;enviamos aenv a las salidas izquierda y derecha
outs aenv, aenv
;fin de la definición del instrumento
endin
</CsInstruments>

;y ahora la partitura:
<CsScore>
;p1 p2 p3 p4 p5
i1 0 5 10000 220
e
</CsScore>
</CsoundSynthesizer>

```

1) Como siempre pegaremos el código en nuestro editor favorito y guardaremos el archivo con la extensión .csd

Compilar con CSound5 GUI y probar a cambiar los valores de los armónicos, sus amplitudes relativas, sus desafinaciones (sin pasarse!) y los parámetros de la envolvente.

2) Con cambios de este tipo crear un sonido nuevo distinto al del ejemplo. Grabar el fichero .csd. Grabar el fichero WAV resultante.

3) Duplicar el archivo .csd y darle el nombre de "ejemplo2aditivaplus.csd" (por ej.). Cambiar los valores de los armónicos, sus amplitudes relativas, sus desafinaciones (sin pasarse!) y los parámetros de la envolvente general. Crear una envolvente distinta para cada uno de los parciales. Enviar los parciales impares a la salida izquierda y los pares a la derecha. Grabar el fichero WAV resultante.

30/10/2007