

CSound: software libre de síntesis sonora

Bienvenidos a esta introducción al lenguaje de síntesis sonora CSound.

El objetivo de este documento es facilitar los primeros pasos con un lenguaje muy eficaz para la generación y el procesado digital de sonido. Con este lenguaje se realizarán las siguientes prácticas de la parte de sonido de la asignatura de Tecnología Multimedia.

El documento ofrece una guía muy detallada de un ejemplo en lenguaje CSound. Las partes que no son indispensables pero que ofrecen información complementaria están encerradas entre corchetes y pueden ser omitidas a discreción.

Dónde conseguirlo

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

La última versión (recomendada) es la 5.07, (justo del 1 de octubre de 2007).

¿Hay documentación o un manual?

Sí:

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

Datos básicos

- CSound es un lenguaje especializado de síntesis sonora. No es un lenguaje de propósito general y por tanto no es adecuado para otras tareas.
- CSound compila ficheros de tipo texto que contienen programas escritos en lenguaje CSound y produce sonido en tiempo real o un fichero de audio para su reproducción posterior.
- Cada fichero en lenguaje CSound contiene las instrucciones necesarias para la generación del sonido resultante y puede ser creado con cualquier editor de textos.
- Normalmente, un fichero CSound contiene instrucciones para la definición de "instrumentos de la orquesta" y una lista o "partitura" de "notas" o eventos sonoros con los datos que interpretará la "orquesta".
- CSound es multi-plataforma y está disponible para todos los sistemas operativos, como Linux y Mac OS X, e incluso para Windows.
- Un fichero en lenguaje CSound puede ser compilado en cualquier plataforma y producirá el mismo resultado sonoro.

PRIMER EJEMPLO

Un fichero CSound vacío sólo contiene una serie de etiquetas XML que marcan las distintas secciones:

Listado: "ejemplo1.csd"

```
<CsoundSynthesizer>
<CsOptions>
</CsOptions>
<CsInstruments>
</CsInstruments>
<CsScore>
</CsScore>
</CsoundSynthesizer>
```

Ahora iremos rellenando una a una las distintas secciones. Documentaremos con comentarios precedidos por ";" cada una de las líneas que añadamos

1) La sección de opciones nos permite añadir banderas y datos para el momento de la compilación. Podemos dejarla en blanco sin problemas.

[En nuestro caso añadiremos simplemente lo siguiente:

```
<CsOptions >
;salida a un archivo WAV de nombre "ejemplo1.wav"
-o ejemplo1.wav
</CsOptions >
```

que indica al compilador CSound que guarde el resultado sonoro en un archivo llamado "ejemplo1.wav" con el formato WAV de Microsoft.]

[Veremos que en muchos casos los ajustes del compilador CSound tendrán prioridad sobre las opciones que figuran en esta sección]

2) La sección de instrumentos contiene la codificación en lenguaje CSound de los procesos matemáticos que vamos a realizar para generar los sonidos. Para ello usaremos variables y *opcodes* que es el nombre que reciben las funciones en CSound.

```
<CsInstruments>
;sample rate
sr = 44100
;control rate
ksmps = 1
;number of channels
nchnls = 1
</CsInstruments>
```

En primer lugar hemos añadido una cabecera que asigna valores a tres variables reservadas (sr, ksmps y nchnls) necesarias para la compilación.

sr significa *sample rate*, frecuencia de muestreo y nos indica cuántas muestras por segundo se generarán. Normalmente se usa el valor de 44100 que corresponde con

la frecuencia de muestreo del CD y que proporciona una buena respuesta en frecuencia*

ksmps significa número de muestras por bloque de control, y da idea de la velocidad de actualización del cálculo de las variables con prefijo k-. [Normalmente y con los ordenadores actuales el valor es 1 que significa que las variables k- se actualizan con cada muestra.]

nchnls significa *number of channels*, número de canales del sonido resultante: 1 = mono, 2 = estéreo, etc.

Ya estamos listos para crear nuestro instrumento generador de sonido.

```
;definición del instrumento 1
instr 1
;asignamos a la variable iamplitude el valor de p4
iamplitude = p4
;asignamos a la variable ifrequency el valor de p5
ifrequency = p5
;asignamos a la variable iphase el valor 0
iphase = 0
;asignamos a la variable asound la salida del opcode oscils
;todos los parámetros de entrada están a la derecha
;las variables de salida a la izquierda
asound oscils iamplitude, ifrequency, iphase
      ;enviamos asound a la salida
      out asound
;fin de la definición del instrumento
endin
```

Finalmente creamos una "partitura" con una sola nota:

```
<CsScore>
;p1 indica que la nota se creará con el instrumento
;p2 indica que la nota empezará para tiempo=0
;p3 indica que la nota durará 5 segundos
;p4 es 10000. El significado depende de la definición del
instrumento 1
;p5 es 440
;p1      p2      p3      p4      p5
i1      0       5       10000  440
;la "e" indica end, fin de partitura
e
</CsScore>
```

El resultado uniendo las distintas secciones es:

Listado: "ejemplo1.csd"

```
<CsoundSynthesizer>

<CsOptions >
;salida a un archivo WAV de nombre "ejemplo1.wav"
-o ejemplo1.wav
</CsOptions >

<CsInstruments>
;sample rate
sr = 44100
;control rate
ksmps = 1
;number of channels
nchnls = 1

;definición del instrumento 1
instr 1
;asignamos a la variable iamplitude el valor de p4
iamplitude = p4
;asignamos a la variable ifrequency el valor de p5
ifrequency = p5
;asignamos a la variable iphase el valor 0
iphase = 0
;asignamos a la variable asound la salida del opcode oscili
asound oscils iamplitude, ifrequency, iphase
    ;enviamos asound a la salida de audio
    out asound
    ;fin de la definición del instrumento 1
endin
</CsInstruments>

<CsScore>
;el valor de p1 indica que la nota se creará con el instrumento 1
;el valor de p2 indica que la nota empezará para tiempo=0
;el valor de p3 indica que la nota durará 5 segundos
;el valor de p4 es 10000
;el valor de p5 es 440
;p1      p2      p3      p4      p5
i1      0       5       10000  440
e
; e indica fin de la partitura
</CsScore>
</CsoundSynthesizer>
```

LAS VARIABLES EN CSOUND

Como se ha visto, todas las variables en CSound van precedidas de una o dos letras. La letra "i" indica que la expresión de la variable se evaluará una sola vez por cada nota que se toque con el instrumento. En nuestro primer ejemplo, las variables iamplitude e ifrequency son inicializadas con los parámetros p4 y p5 procedentes de la partitura.

Las variables que empiezan por "a" se recalculan con la frecuencia marcada por la variable de sistema sr, es decir tantas veces por segundo como indique su valor. En nuestro ejemplo la variable asound será recalculada 44100 veces cada segundo y su valor se enviará a la salida 44100 veces.

Las variables que empiezan por "k" se usan para valores que no requieren ser actualizados tan a menudo como las variables con "a" pero que necesitan modificarse durante la duración de la nota. Un ejemplo sería el cálculo de la envolvente pero esto lo veremos posteriormente.

[Las variables que empiezan por "gi", "gk" o "ga" son variables globales compartidas por todos los instrumentos].

LA PARTITURA

Como hemos visto la partitura contiene la información de las notas que se compilarán. En nuestro primer ejemplo sólo tenemos una nota:

```
;p1      p2      p3      p4      p5
i1       0       5       10000  440
;la "e" indica fin de partitura
e
```

Vemos que esta nota contiene 5 parámetros.

Los tres primeros tienen un significado fijo en CSound:

el valor de p1 siempre indica el instrumento con el que se ejecutará la nota,

el valor de p2 siempre indica el tiempo al que empezará la nota,

el valor de p3 siempre indica la duración en segundos.

Nuestra nota contiene dos parámetros adicionales, p4 y p5 que (tradicionalmente) el instrumento 1 asigna a la amplitud y a la frecuencia, pero no hay ninguna razón para que no los asigne a otro propósito.

Vamos a añadir una nota adicional a la partitura:

```
;p1      p2      p3      p4      p5
i1       0       5       10000  440
i1       5       5       5000   220
e
```

La segunda nota se interpretará con el instrumento 1, a partir del segundo 5, o sea justo después de la primera nota, durará 5 segundos y tendrá una amplitud de 5000 y una frecuencia de 220.

CSOUND ES UN PROGRAMA Y UN LENGUAJE

CSound es tanto el programa que compila los ficheros CSound como el nombre del lenguaje de los ficheros CSound. CSound es un programa open-source y como tal su código fuente está disponible libremente. Pero compilar el CSound es otra cosa... Nosotros siempre usaremos las versiones ya compiladas para el sistema operativo Microsoft Windows o para Linux.

¿COMPILAR O COMPILAR?

¿Por qué decimos que CSound compila? CSound produce ficheros de audio (aunque estos se reproduzcan en tiempo real) como resultado de la compilación de código escrito en lenguaje CSound.

Pensemos por ejemplo en el Acrobat Reader, o en un cualquier otro visor de ficheros PDF, que compila ficheros escritos en lenguaje PDF (un subconjunto del lenguaje Postscript) y ofrece como resultado ficheros bitmap (de mapa de puntos, color o blanco y negro) visualizables en pantalla e imprimibles por impresora.

Igualmente GraphViz compila ficheros escritos en lenguaje DOT para producir hermosos grafos (por ejemplo en formato PDF...)

Como ya dijimos al principio, CSound no es un programa de propósito general: Ver http://en.wikipedia.org/wiki/Domain-specific_programming_language

PRÁCTICA 2.-Introducción

- 1) Descargar e instalar el programa CSound.
- 2) Copiar en cualquier editor de texto el contenido del "ejemplo1.csd". Guardar con la extensión .csd
- 3) Ejecutar el programa CSound 5 GUI. Este programa proporciona una interface gráfica muy simple al motor de CSound, que se ejecuta en la terminal.
- 4) Hacer clic en el botón marcado con ... arriba a la izquierda.

Esto nos permite cargar un fichero .csd.

Habilitar la entrada/salida en tiempo real del sonido (Realtime I/O)

Hacer clic en el icono de sintetizar (play)



- 5) Deshabilitar la entrada/salida en tiempo real del sonido (Realtime I/O). Sintetizar de nuevo y comprobar que se ha generado un fichero de sonido (normalmente de tipo Microsoft WAVE). Abrirlo en el Audacity y reproducirlo.
- 6) Crear 3 nuevos documentos .csd que contengan modificaciones o nuevas creaciones. Sugerencias:
 - Varias notas al mismo tiempo con distintas frecuencias muy cercanas
 - Varias notas al mismo tiempo con frecuencias que guarden proporciones armónicas (frecuencia = f , $2*f$, $3*f$, etc) con distintas amplitudes. Abrir los resultados en el Audacity y realizar el análisis espectral.
 - Lo mismo pero entrando las notas paulatinamente en tiempos distintos y acabando todas juntas. También se pueden usar pausas entre las notas
 - Crear los sonidos límites de la audición humana
 - Crear una escala de 7 notas (Do, Re, Mi, Fa, Sol, La, Si) o cromática [sólo para alumnos con conocimientos musicales].

Pista: la proporción entre dos sonidos de frecuencia f_1 y f_2 separados por un semitono es la raíz doceava de 2 = 1,059463094359. Por ejemplo $La_4 = 440$ Hz, $La\#_4 = 1.06 * 440 = 466$ Hz
- 7) Sintetizar los sonidos correspondientes y guardarlos en archivos WAVE (wav).