

Información

Ejercicio 4.8: if - then - else

En este ejercicio vamos a ver cómo se usa este tipo de estructura de control del flujo de ejecución.

En Csound, el controlador del flujo de ejecución condicional **if** se escribe así:

```
if condición then      ; la condición puede ir o no entre paréntesis.
    sentencias si se cumple la condición
else                    ; o bien puede ser otro elseif condición then
    sentencias si no se cumple la primera condición ; o si se cumple la segunda
endif
```

donde la parte del “**else**” puede no usarse. Para escribir la condición podemos asumir que la sintaxis para ello del lenguaje C (C++ o Java) es válida aquí.

- Haz una copia del proyecto del ejercicio anterior y llama al nuevo `ej4.8.csd`.

<CsOptions> (parámetros de compilación)

- Cambia el fichero de salida para que ahora sea `ej4.8.wav`.

</CsInstruments> (orquesta)

- De los parámetros globales, cambiaremos el valor del número de canales para que ahora sea una salida estéreo: **nchnls** = 2
- También se cambiará el máximo valor codificable (**0dbfs**) para que ahora tengamos señales de amplitud normalizada: **0dbfs** = 1

Vamos a usar los canales estéreo, generando ruido rosa con el operador **pinkish**, según se seleccione desde la partitura, mediante el **if-then-else**.

- Cambia el nombre al instrumento para que sea el **Canal**.
- Quita la asignación a frecuencia de **p5**. Ahora en **p5** se pasará desde la partitura el canal del estéreo a utilizar, con `iCanal = p5`, según el siguiente criterio:
 - 0 : sonido solo por el canal izquierdo,
 - 1 : sonido solo por el canal derecho,
 - 2 : sonido igual por ambos canales.
- Quita también el generador de ondas sinusoidales del ejercicio anterior.
- El valor de `iCanal` se usará para seleccionar el canal por el que emitir el ruido, mediante **if-then-elseif- ... -endif**. Si se pasa un número diferente, el instrumento imprimirá un mensaje de error en la consola mediante **prints “ERROR!\n”**
- En cada rama del **if** deberás utilizar 2 señales de audio (variables de tipo a-), una para cada canal:
 - La variable del canal que lleve señal sonora se genera con **variable_señal pinkish iAmp**.
 - La variable del canal que no lleve señal debes asignarle un cero (**variable_señal = 0**).
 - En el caso de la señal estéreo, ambos canales llevarán el ruido.
 - En el caso del error, a ambos canales se les asignará cero.
- La salida, como ahora es estéreo, hay que cambiarla por:

```
outs variable_usada_para_el_izquierdo , variable_usada_para_el_derecho
```

<CsScore> (partitura)

- Sustituye la orden de tempo para que sea ahora de 50 BPM
- Sustituye las notas de la partitura, por 4 activaciones del instrumento **"Canal"** de 1 tiempo cada una, que empiecen en los tiempos 0, 2, 4 y 6, de amplitud -5 dBFS, y que seleccionen, respectivamente, los canales 0, 1, 2 y 3.

Funcionamiento:

Compila y prueba el funcionamiento. Verifica en la consola que

```
overall samples out of range:      0
```

Y que se oyen los 3 ruidos por los diferentes canales y al final se escribe **ERROR!** en la consola.

