



Interfacce grafiche in CsoundAV

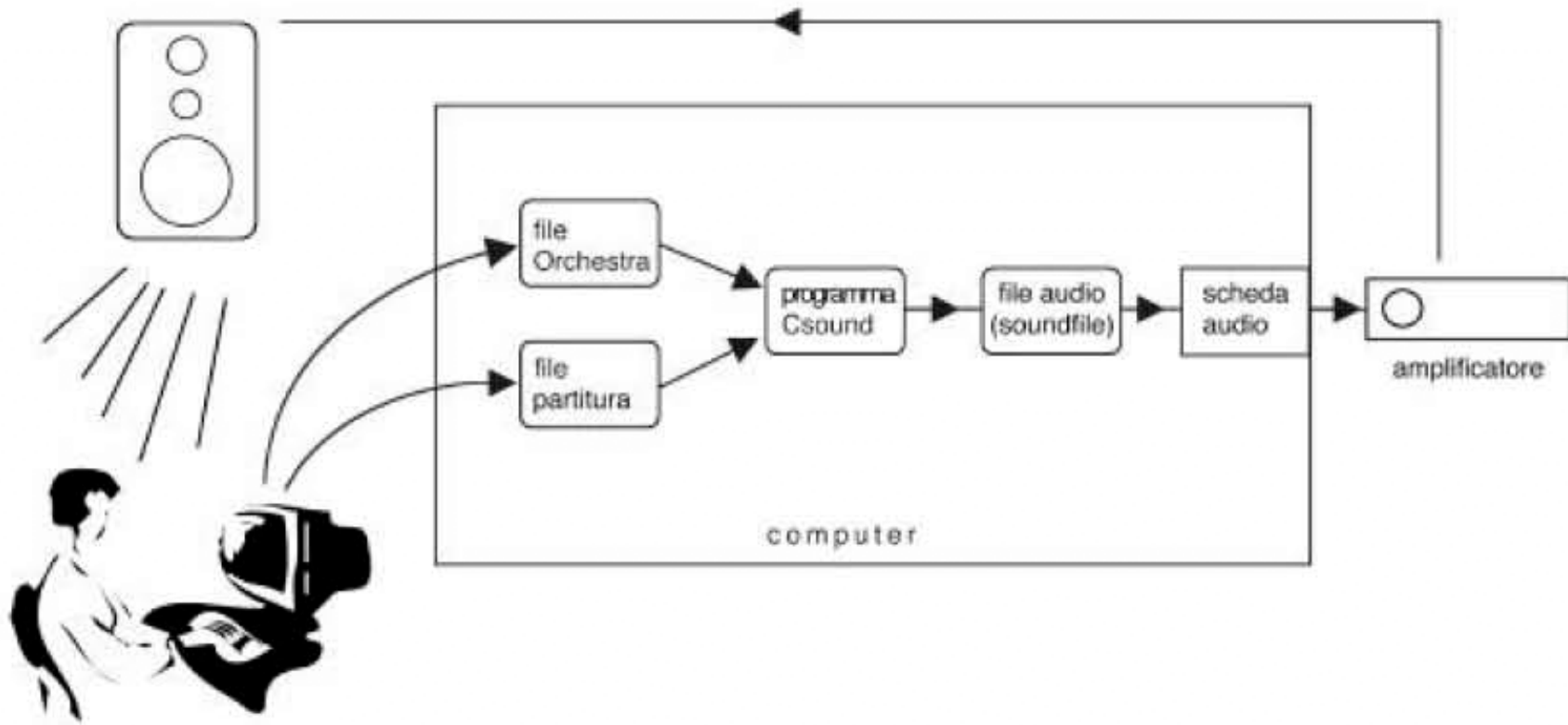
Alessandro Petrolati

www.alessandro-petrolati.com

L.E.M.S.

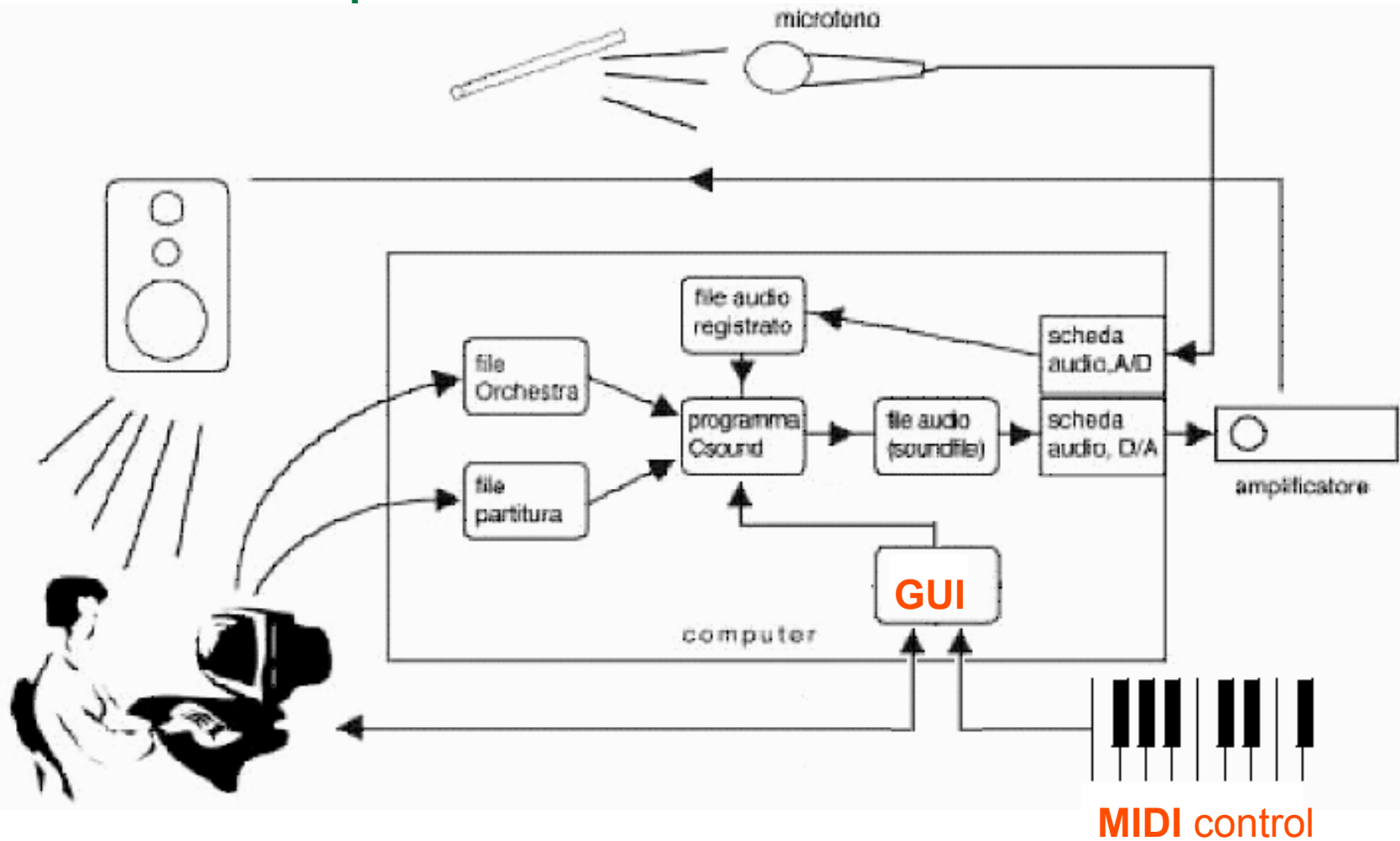
marzo/aprile 2005

Compilazione file audio (non real time)



TempoCompilazione \geq TempoComposizione

Real time performance



TempoCompilazione < TempoComposizione

Csound history

Music 1 scritto da Max Mathews nel 1957 su un calcolatore [IBM 704](#) ai **Bell Labs**.

Music 2 del 1958 permetteva la generazione di qualsiasi forma d'onda.

Music 3 del 1959 scritto per la nuova generazione dell' [IBM 7094](#)

Music 4 del 1960 sempre Max Mathews alla Princeton e Stanford University, introdotto il generatore d'inviluppo (envlp)

Music4B 1966-67 trascritto in BEFAP assembler, introdotti i filtri risonanti.

Music4BF per la serie [IBM 360](#), poiché quest'ultima rendeva incompatibile la codifica BEFAP, Godfrey lo trascrisse in fortran, più lenta ma ovunque portabile.

Music 5 del 1966 sempre di Max Mathews.

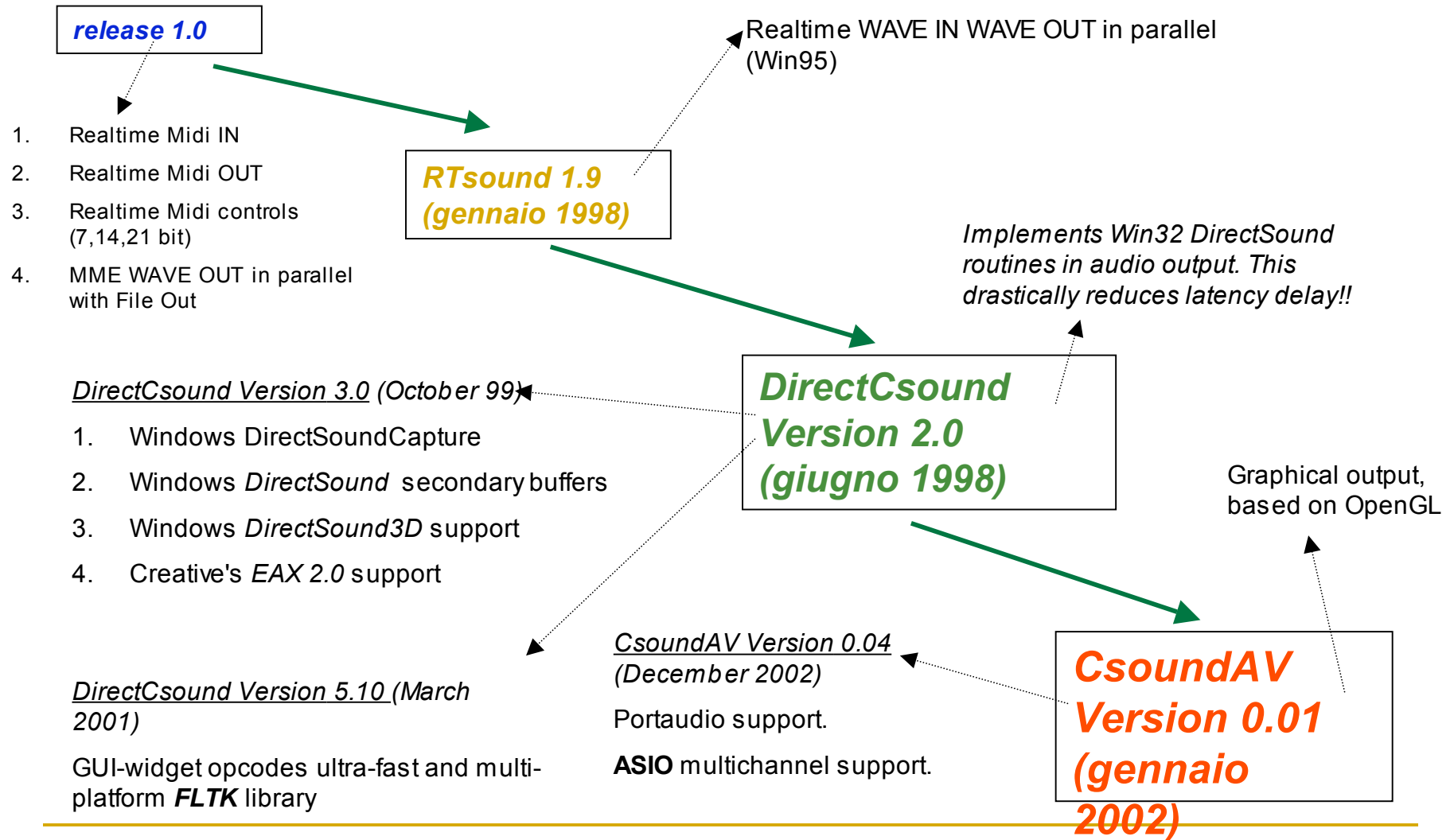
Music 360 del 1968 sviluppato da Barry Vercoe del MIT (Massachusetts Institute of Technology)

Music 10 da John Chowning e James Moorer all'università di Stanford.

Music 11 del 1973 di Barry Vercoe al M.I.T. (Massachusetts Institute of Technology)

CSound è stato scritto nel 1986 da Barry Vercoe con l'aiuto di Kevin Peterson, Alan Delespinase, Bill Gardner, Dan Ellis e Paris Smaragdis, la codifica è stata portata dall'assembler del [PDP-11](#) al C di Unix.

Distribuzione G. Maldonado



Csound in real time

- **-+Pn** Portaudio **output** device (ASIO, MME and DirectSound devices)
- **-+In** (uppercase 'i') Portaudio **input** device
- **-+Kn** Enable **MIDI In**
- **-+Qn** Enable **MIDI Out**
- **-f** float sound samples
- **-+e** Allow **any output sample rate** (for cards that support this feature)
- **-+O** (uppercase letter) Suppress all printf for better realtime performance.
- **-mn** tty message level (0=suppress all)
- **-d** suppress all displays

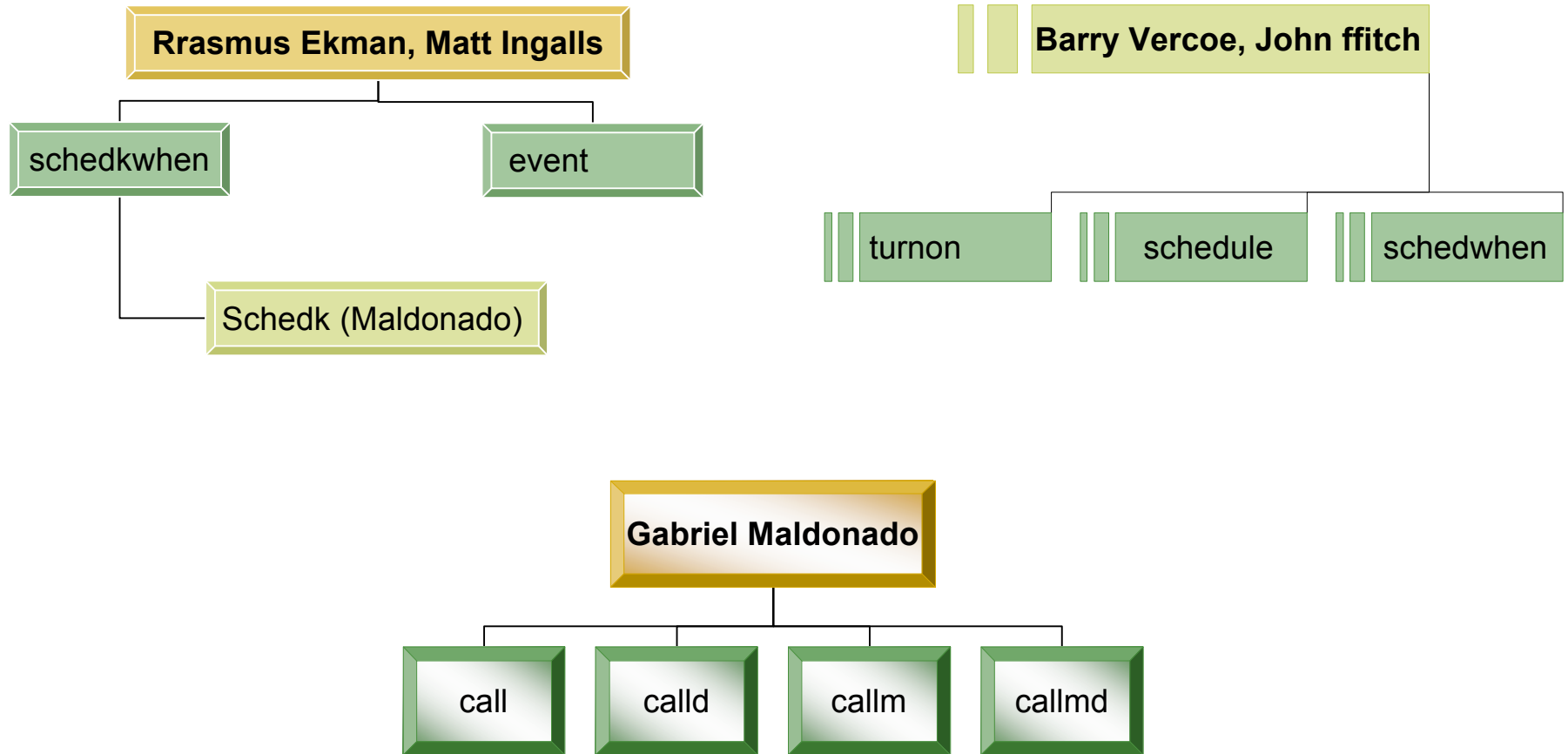
Esempi

-+P -+I -m0 -+O -d ;abilita audio IN/OUT e sopprime messaggi grafici

-+e -P99 -+K4 -+Q5 ;disabilita l'audio e abilita MIDI IN/OUT sui device 4 e 5

N.B. Se non viene esplicitato nessun numero audio/midi, una finestra compare al momento dell'avvio.

Orchestra schedule instrument



Orchestra trigger

- **kr trigger** ksig, kthreshold, kmode
- **ktrig metro** kfreq
- **ktrig changed** kvar1 [, kvar2,..., kvarN]

Alternativa:

reset:

iTime = i(*kTrigTime*)

timeout 0, iTime, contin

reinit reset

contin:

call instrno, iactime, idur [, ip4, ip5, ip6, ip7, ... , ipn]

Turning off orchestra instrument

- **turnoff**
- **turnoffk** ktrig

Alternativa:

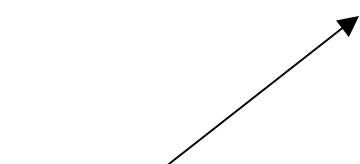
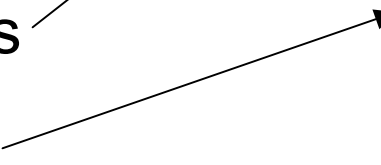

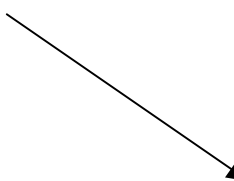
iDur **init** *p3*

timeout 0, iDur,contin

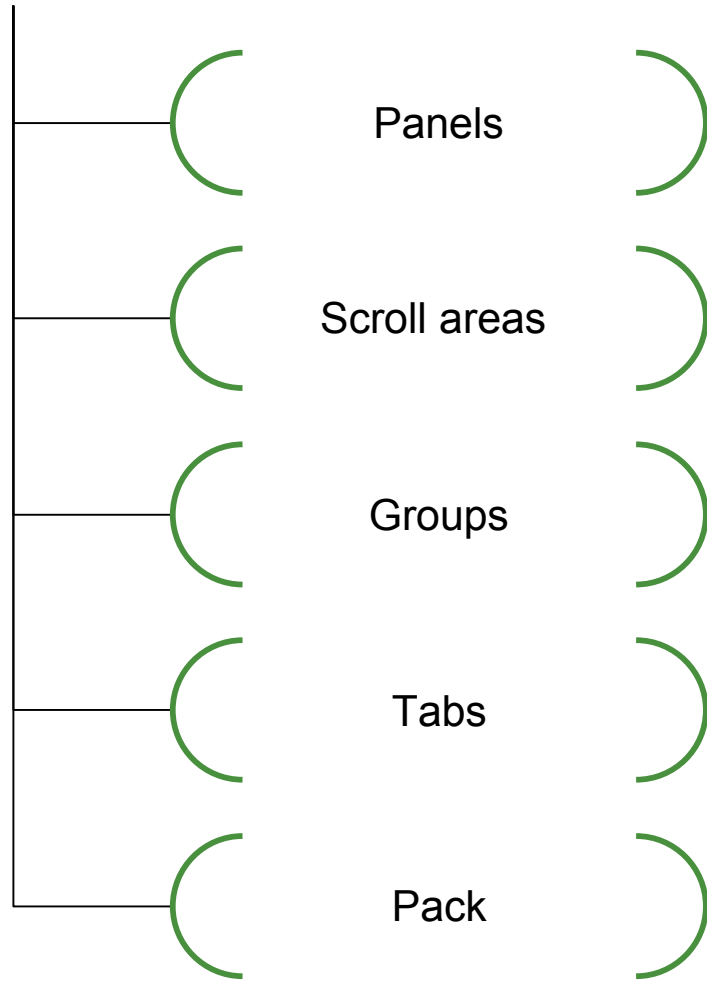
turnoff

contin:

FLTK (Fast Light Tool Kit)

- ✓ Containers  Sono Widgets che contengono altri controlli
es: Altri pannelli, Finestre etc.
 - ✓ Valuators  Rimandano variabili di tipo “k” per il controllo
dei parametri di sintesi in tempo reale, da
usare dentro gli strumenti Csound
 - ✓ Other widgets  Implementano funzioni particolari
 - ✓ Appearance modifier  Cambiano l’aspetto grafico dei controlli
-

Containers



SimplePanel.csd



simpleScroll.csd

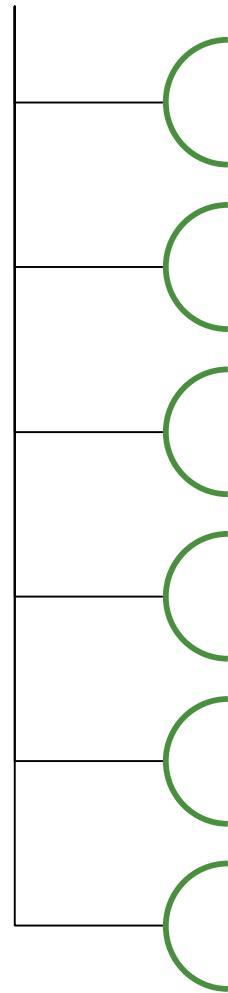


simpleTabs.csd



Pack.csd

Valuators



Slider

Knob

Roller

Text

Joystick

Counter



Esempi.csd



simpleSlider.csd



SimpleKnob.csd



simpleRoller.csd



simpleTextField.csd

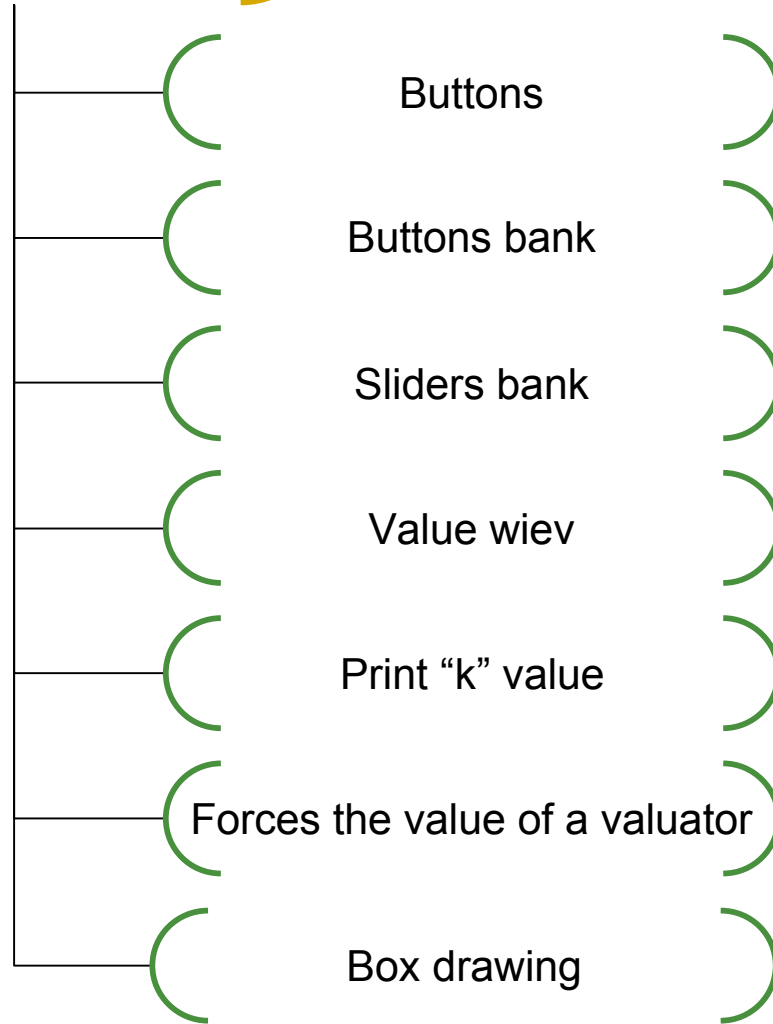


simpleJoystick.csd



SimpleCounter.csd

Other widgets



simpleButton.csd



simpleButtonBank.csd

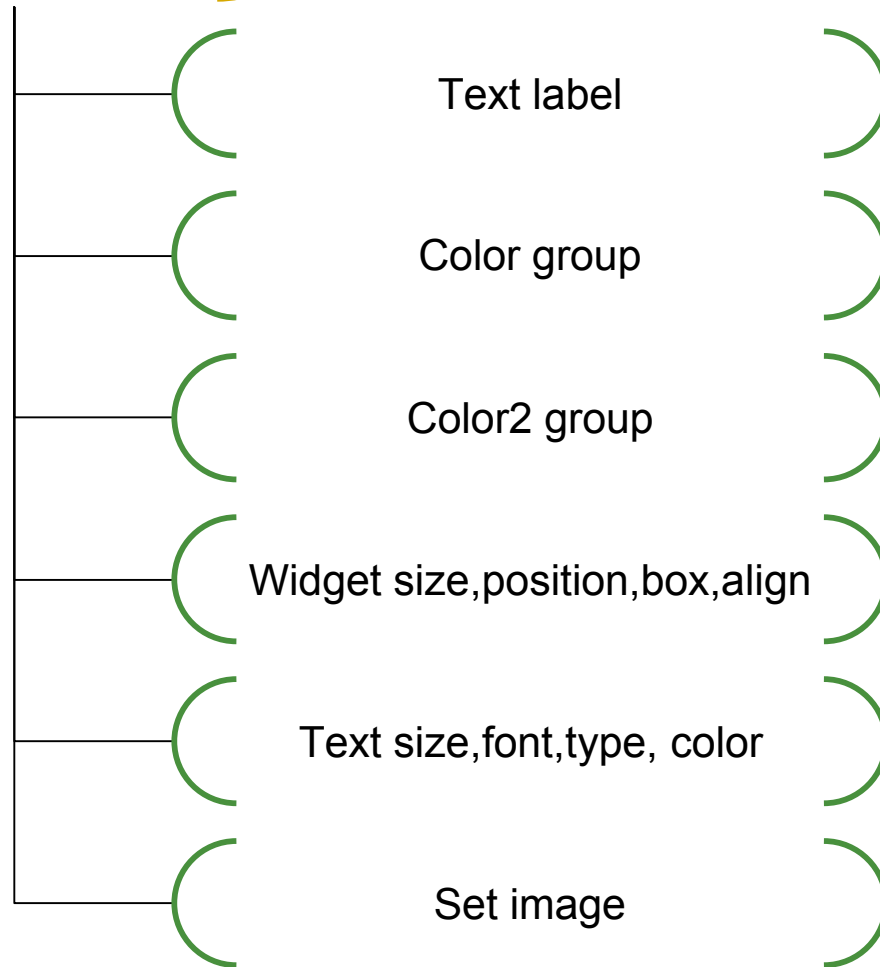


simpleSlidBnk.csd

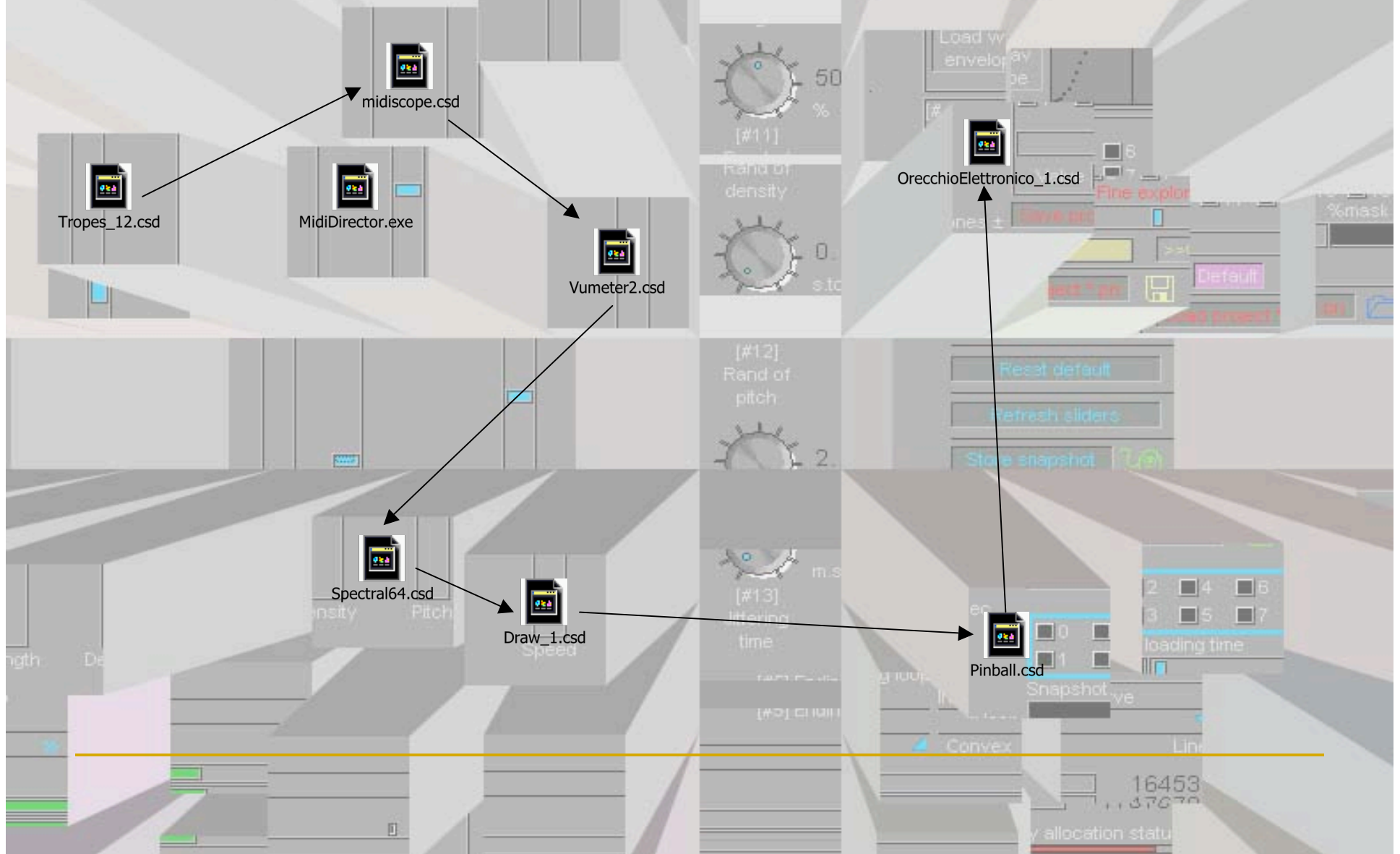


Fprintk.csd

Appearance modifier



GUI examples



Opcode list_1

- `FLpanel` "label", iwidth, iheight, ix, iy [, iborder]
 - `FLpanel_end` *oppure* `FLpanelEnd`
 - `FLscroll` iwidth, iheight, ix, iy
 - `FLscroll_end` *oppure* `FLscrollEnd`
 - `FLtabs` iwidth, iheight, ix, iy
 - `FLtabs_end` *oppure* `FLtabsEnd`
 - `FLgroup` "label", iwidth, iheight, ix, iy [, iborder]
 - `FLgroup_end` *oppure* `FLgroupEnd`
 - `FLpack` iwidth,iheight,ix,iy,itype,isspace,iborder **INCOMPLETO** in Rasmus Ekman Win Help
 - `FLpack_end` *oppure* `FLpackEnd`
 - `FLrun`
 - `FLupdate` **NON documentato**
 - kout, ihandle `FLslider` "label", imin, imax, iexp, itype, idisp, iwidth, iheight, ix, iy
 - kout, ihandle `FLknob` "label", imin, imax, iexp, itype, idisp, iwidth, ix, iy
 - kout, ihandle `FLroller` "label", imin, imax, istep, iexp, itype, idisp, iwidth, iheight, ix, iy
 - kout, ihandle `FLtext` "label", imin, imax, istep, itype, iwidth, iheight, ix, iy
 - koutx, kouty, ihandlex, ihandley `FLjoy` "label", iminx, imaxx, iminy, imaxy, iexpx, iexpy, idispx, idispy, iwidth, iheight, ix, iy
 - kout, ihandle `FLcount` "label", imin, imax, istep1, istep2, itype, iwidth, iheight, ix, iy, iopcode [, kp1, kp2, kp3,, kpN]
 - kout, ihandle `FLbutton` "label", ion, ioff, itype, iwidth, iheight, ix, iy, iopcode [,kp1, kp2, kp3, kp4, kp5,, kpN]
 - kout, ihandle `FLbutBank` itype, inumx, inumy, iwidth, iheight, ix, iy, iopcode [,kp1, kp2, kp3, kp4, kp5,, kpN]
 - ihandle `FLbox` "label", itype, ifont, isize, iwidth, iheight, ix, iy
 - ihandle `FLvalue` "label", iwidth, iheight, ix, iy
 - inumSnap, inumVal `FLsetsnap` index [, ifn]
 - inumSnap `FLgetsnap` index
 - `FLsavesnap` "filename"
 - `FLloadsnap` "filename"
-

Opcode list_2

- FLprintk itime, kval, idisp
- FLprintk2 kval, idisp
- FLsetVal_i ivalue, ihandle
- FLsetVal ktrig, kvalue, ihandle
- FLlabel isize, ifont, ialign, ired, igreen, iblue
- FLcolor ired1, igreen1, iblue1, ired2, igreen2, iblue2
- FLcolor2 ired, igreen, iblue
- FLsetColor ired, igreen, iblue, ihandle
- FLsetColor2 ired, igreen, iblue, ihandle
- FLsetSize iwidth, iheight, ihandle
- FLsetPosition ix, iy, ihandle
- FLsetBox itype, ihandle
- FLsetAlign ialign, ihandle
- FLhide ihandle
- FLshow ihandle
- FLsetTextSize isize, ihandle
- FLsetTextColor ired, igreen, iblue, ihandle
- FLsetFont ifont, ihandle
- FLsetTextType itype, ihandle
- FLsetText "itext", ihandle
- FLtextSize isize
- FLsetImage imageHandle, ihandle [, iflag]
- FLsetOverlay ktrig, kx, ky, kwidth, kheight, ihandle
- **Non documentati**
- "BROWSE_FILE" per gli opcode che prevedono il parametro "filename"
- kr max_k asig, ktrig, 0 trova il valore d'involuppo in un segnale vedi [follow](#). (L'ultimo paramentro deve essere zero poichè non documentabile)

FL panel, tabs, group syntax_1

```
<CsoundSynthesizer>
<CsOptions>
;contiene le Csound Flags
</CsOptions>
<CsInstruments>
;header opzionale
sr=44100
kr=441
ksmps=100
nchnls=1

    FLpanel "Pannello",400,250                ;crea un pannello principale che contiene tutto
    FLtabs 400-10,250-10, 5,5                ;le tabs possono contenere più gruppi

    FLgroup "Tab_1",400-20,250-60, 50,30,0 ;primo gruppo
;contiene l'interfaccia grafica Tab_1
    FLgroup_end                               ;fine primo gruppo

    FLgroup "Tab_2",400-20,250-60, 10,30,0 ;secondo gruppo
;contiene l'interfaccia grafica Tab_2
    FLgroup_end                               ;fine secondo gruppo

    FLgroup "Tab_3",400-20,250-60, 10,30,0 ;terzo gruppo
;contiene l'interfaccia grafica Tab_3
    FLgroup_end                               ;fine terzo gruppo

    FLtabs_end                               ;fine delle tabs che contengono i gruppi
    FLpanel_end                              ;termina il pannello generale
    FLrun                                    ;inizializza la GUI

    instr 1
;contiene la codifica Csound Orchestra
    endin

</CsInstruments>
<CsScore>
;contiene lo score
</CsScore>
```

Diagram illustrating the structure of the FL panel syntax with arrows pointing to the corresponding code lines:

- ← crea un pannello principale che contiene tutto (points to `FLpanel`)
- ← le tabs possono contenere più gruppi (points to `FLtabs`)
- ← primo gruppo (points to `FLgroup "Tab_1"`)
- ← fine primo gruppo (points to `FLgroup_end`)
- ← secondo gruppo (points to `FLgroup "Tab_2"`)
- ← fine secondo gruppo (points to `FLgroup_end`)
- ← terzo gruppo (points to `FLgroup "Tab_3"`)
- ← fine terzo gruppo (points to `FLgroup_end`)
- ← fine delle tabs che contengono i gruppi (points to `FLtabs_end`)
- ← termina il pannello generale (points to `FLpanel_end`)
- ← inizializza la GUI (points to `FLrun`)

FL scroll syntax_2

```
<CsoundSynthesizer>
<CsoundOptions>
;Csound Flags
</CsoundOptions>
<CsoundInstruments>
kr=44100
ksmps=4410
ksmps = 10
nchnls=1
```

```
FLpanel "This Panel contains big Knob and scroll bars",300,300
FLscroll 300,300,0,0,1
;GUI
FLscroll_end
FLpanel_end
FLrun
instr 1
;Orchestra code
endin
```

```
</CsoundInstruments>
<CsoundScore>
;Score
</CsoundScore>
</CsoundSynthesizer>
```



Lo scroll si può usare anche dentro i gruppi

FLpack

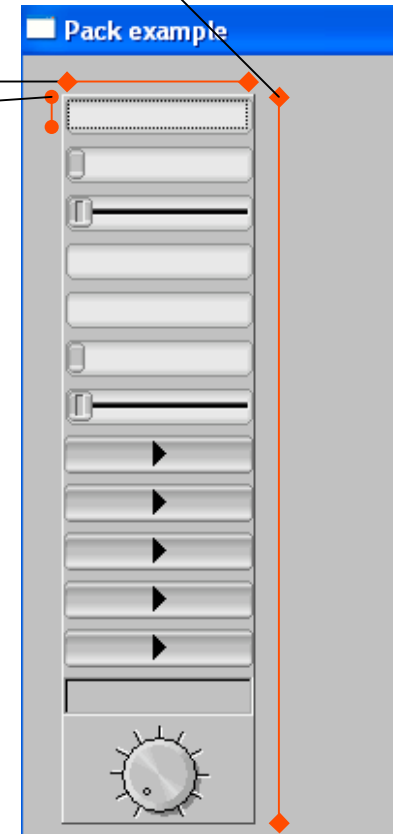
Type 0 per l'impacchettamento verticale;

Type 1 per orizzontale

```
FLpanel "Pack example",450,500,100,100
; iwidth, iheight,ix, iy,itype, ispace, iborder
FLpack 100, 0, 20, 20, 0, 10, 7

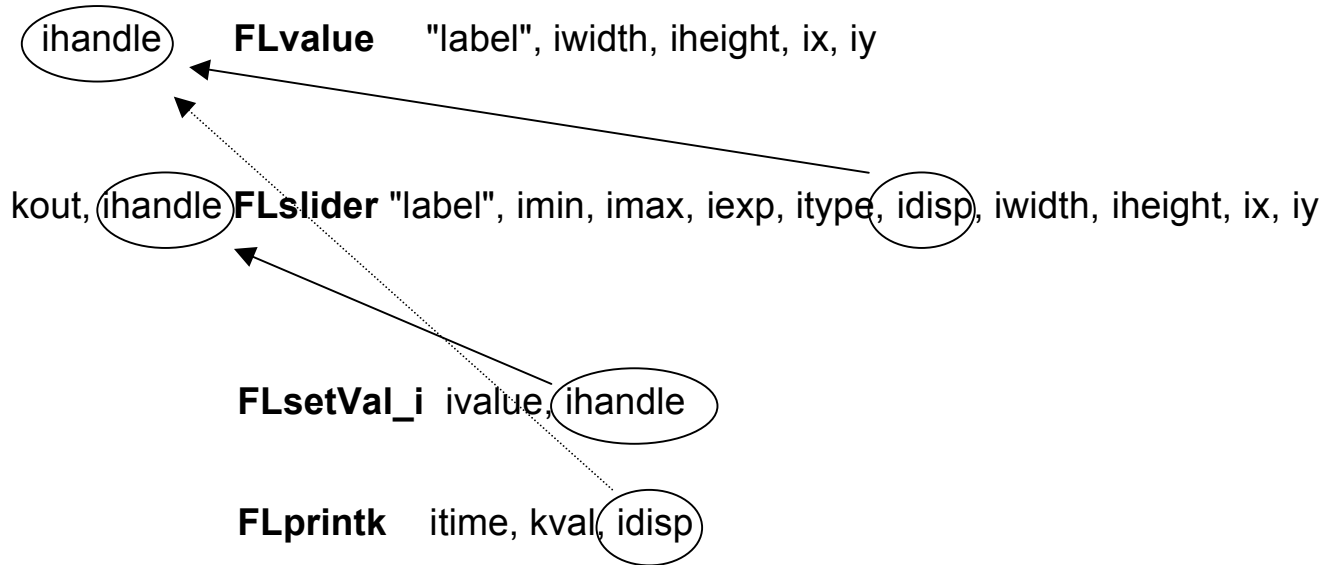
gk1,ihs1 FLslider "1", 500, 1000, -1 ,1, -1, 0,20,0,0
gk2,ihs2 FLslider "2", 300, 5000, -1 ,3, -1, 0,20,0,0
gk3,ihs3 FLslider "3", 350, 1000, -1 ,5, -1, 0,20,0,0
gk4,ihs4 FLslider "4", 250, 5000, -1 ,11, -1, 0,20,0,0
gk5,ihs5 FLslider "5", 220, 8000, -1 ,1, -1, 0,20,0,0
gk6,ihs6 FLslider "6", 1, 5000, -1 ,13, -1, 0,20,0,0
gk7,ihs7 FLslider "7", 870, 5000, -1 ,15, -1, 0,20,0,0
kplay_1, gihST1 FLbutton "@>",1, 0, 1, 0,20,0,0,+105,1,0,3
kplay_2, gihST2 FLbutton "@>",1, 0, 1, 0,20,0,0,+105,1,0,3
kplay_3, gihST3 FLbutton "@>",1, 0, 1, 0,20,0,0,+105,1,0,3
kplay_4, gihST4 FLbutton "@>",1, 0, 1, 0,20,0,0,+105,1,0,3
kplay_5, gihST5 FLbutton "@>",1, 0, 1, 0,20,0,0,+105,1,0,3
gih2 FLvalue " ", 0,20,0,0
gkwidth,ginit2 FLknob "[#1] % Width", 0,1, 0, 1, gih2, 50,0,0,0

FLpack_end
FLpanel_end
FLrun
```



L'altezza del "pacco" dipende dall'altezza dei widgets e dallo spazio che li separa

FL Csound syntax_3



kout, ihandle **FLbutton** "label", ion, ioff, itype, iwidth, iheight, ix, iy, iopcode [,kp1, kp2,, kpN]

+105 abilitato

-105 disabilitato

Slider Bank

```
giElem      init    32
giOutTab    ftgen   1,0,giElem,-2,0
giTabHandle ftgen   4,0,giElem,-5, 0.001,9,1,9,0.001,giElem-18,0.001

        FLpanel "This Panel contains a Slider Bank",400,600
        FLslidBnk "@ ", giElem, giOutTab,300,500,30,30
gihandle FLslidBnkGetHandle
        FLslidBnkSet gihandle, giTabHandle
        FLpanel_end
        FLrun

instr 1
itime init .1
istartSlid init 0
reset:
istartSlid init istartSlid +1
istartSlid init (istartSlid < giElem ? istartSlid : 0)
timeout 0,itime,contin
reinit reset
contin:
FLslidBnkSet gihandle, giTabHandle, 0, istartSlid
endin
```

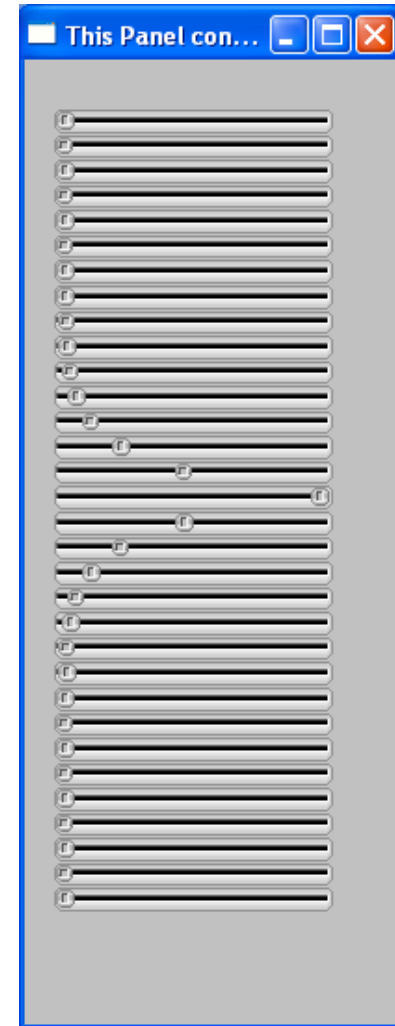
Numero degli
sliders

Inizializza gli slider con i valori
contenuti in giTabHandle

Usato dentro uno strumento può aggiornare i valori
degli sliders ogni volta che viene inizializzato o
reinizializzato



simpleSlidBnk.csd



N.B. *giElem* in questo caso deve essere potenza di due, poichè la variabile è usata come gandezza delle tabelle.

Store/Get/Save/load snapshots bank

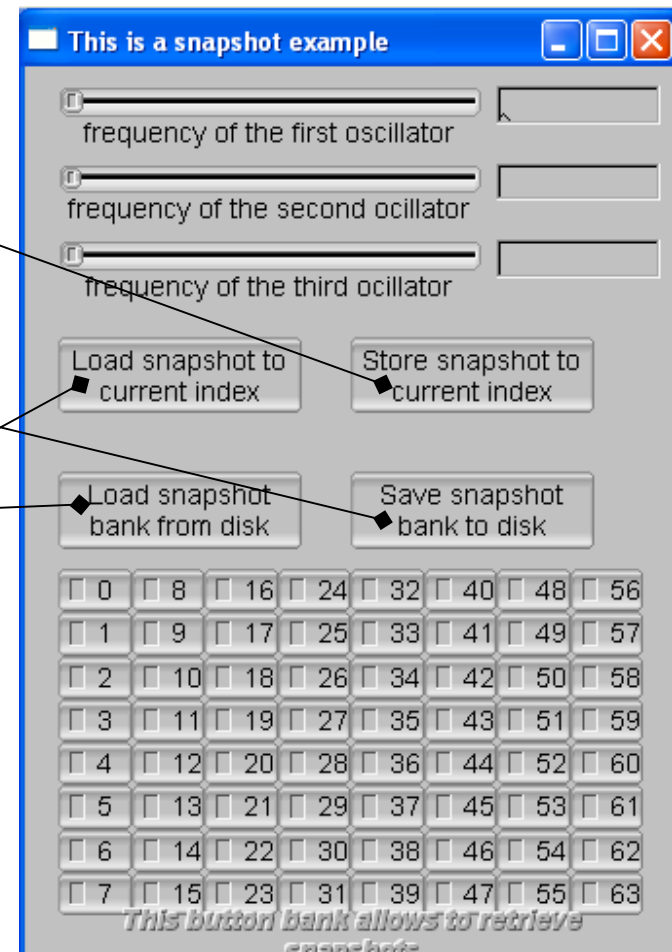
```
giSnapFun ftgen 2,0,256,-2,0
giNumWid init 3
```

```
instr 2 ;store snapshots into gen function
tabw_i i(gk1), 0+(i(gkGet)*giNumWid), giSnapFun
tabw_i i(gk2), 1+(i(gkGet)*giNumWid), giSnapFun
tabw_i i(gk3), 2+(i(gkGet)*giNumWid), giSnapFun
;inumval inumval FLsetsnap i(gkStore)
endin

instr 5 ;save function into bank file txt
ftsave "BROWSE_FILE", 1,giSnapFun
; FLsavesnap "Snapshots.snap"
endin

instr 6 ;load bank from file txt
ftload "BROWSE_FILE", 1,giSnapFun
; FLloadsnap "Snapshots.snap"
endin

instr 7 ;get snap value
ival1 tab_i 0+(i(gkGet)*giNumWid),giSnapFun
ival2 tab_i 1+(i(gkGet)*giNumWid),giSnapFun
ival3 tab_i 2+(i(gkGet)*giNumWid),giSnapFun
FLsetVal_i ival1, gih1
FLsetVal_i ival2, gih2
FLsetVal_i ival3, gih3
;inumel FLgetsnap i(gkGet)
endin
```



Time transition from two snapshots_1

```
giUp sliders      init      12      ;Sliders graphic update
giend             init      1
gkinstrLoad      init      7
gispace          init      4
giNumSlid        init      3
#define DEFAULT # 80,80,80,80 #
gisna0 ftgen 200, 0, gispace, -2, $DEFAULT
gisna1 ftgen 201, 0, gispace, -2, $DEFAULT
gisna2 ftgen 202, 0, gispace, -2, $DEFAULT
gisna3 ftgen 203, 0, gispace, -2, $DEFAULT
gisna4 ftgen 204, 0, gispace, -2, $DEFAULT
gisna5 ftgen 205, 0, gispace, -2, $DEFAULT
gisna6 ftgen 206, 0, gispace, -2, $DEFAULT
gisna7 ftgen 207, 0, gispace, -2, $DEFAULT
gisna8 ftgen 208, 0, gispace, -2, $DEFAULT
gisna9 ftgen 209, 0, gispace, -2, $DEFAULT
gisna10 ftgen 210, 0, gispace, -2, $DEFAULT
gisna11 ftgen 211, 0, gispace, -2, $DEFAULT
gisna12 ftgen 212, 0, gispace, -2, $DEFAULT
gisna13 ftgen 213, 0, gispace, -2, $DEFAULT
gisna14 ftgen 214, 0, gispace, -2, $DEFAULT
gisna15 ftgen 215, 0, gispace, -2, $DEFAULT
giTrfuns ftgen 198,0,2,-2,gisna0,gisna1 ;Temporized Transition
giTrans ftgen 199,0,gispace,-2,0 ;contents of f199 don't matter
giTEMP ftgen 260,0,gispace,-2,0
```



Transition.csd

Time transition from two snapshots_2

```

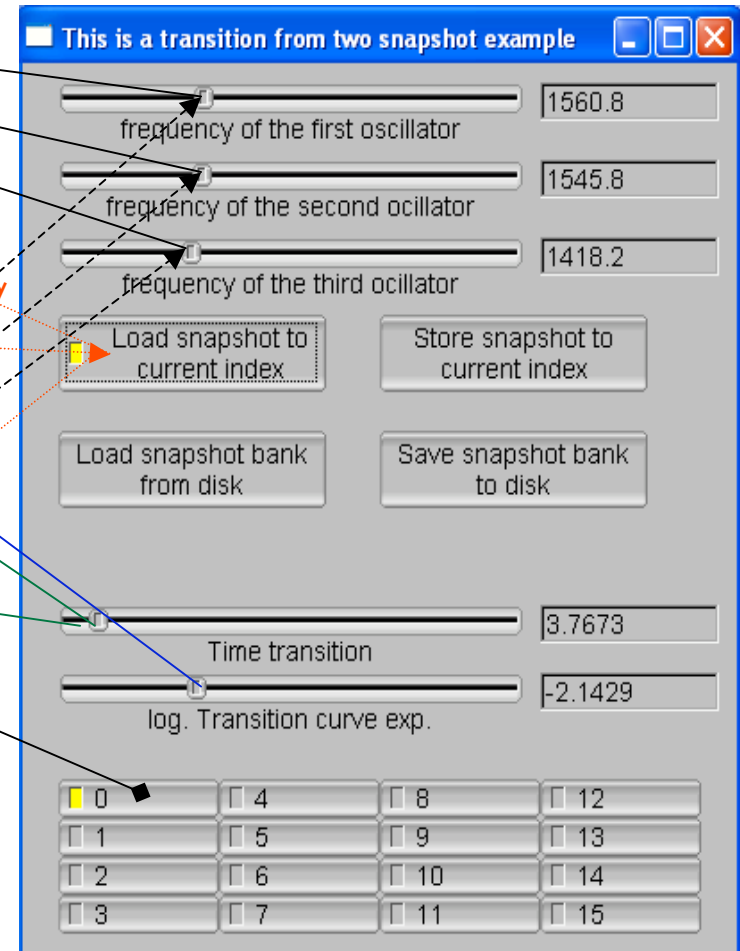
instr 7 ;Transition from two snapshots
gkistrLoad = 8
iType  init i(gkType)
vtabwi 0,giTEMP,i(gk1),i(gk2),i(gk3)
turnoffk gkTrans
timeout 0, i(gkTIME),contin
igoto contin
reinit reset

reset:
FLsetVal_i 1, gihOn
turnoff
rigoto end

contin:
tabw_i giTEMP, 0,giTrfuns
tabw_i gisna0+i(gkGet), 1,giTrfuns
kTr transeg 0, i(gkTIME), iType, 1
ktrig metro giUpsliders
ftmorf kTr, giTrfuns, giTrans
vtabk 0,giTrans,gk1,gk2,gk3
FLsetVal ktrig,gk1,gih1
FLsetVal ktrig,gk2,gih2
FLsetVal ktrig,gk3,gih3

end:
endin
instr 8 ;resetta il numero di strumento transizione
gkinstrLoad init 7
turnoff
endin

```



Call instr 7 when checked

Unchecked button

Call instr 8

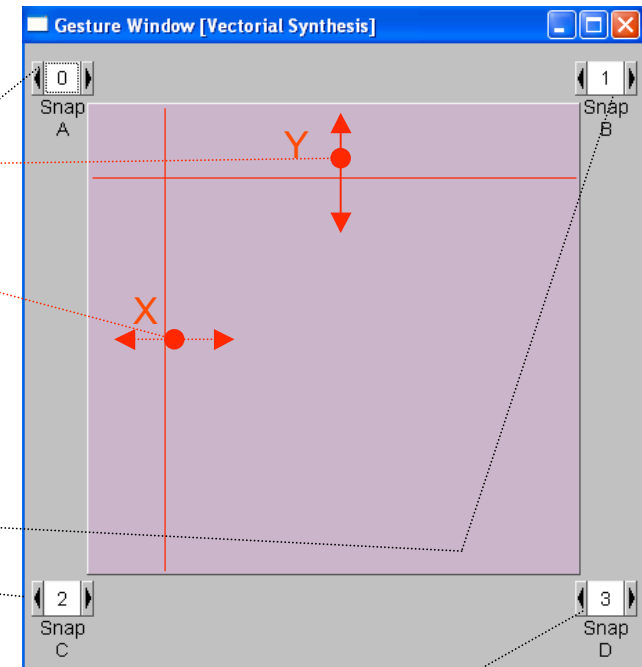
Vectorial synthesis

```
giSnapAB ftgen 190,0,2,-2,gisna0 ,gisna1 ;contiene i numeri di snapshots A e B (snap0 - snap1)
giSnapCD ftgen 191,0,2,-2,gisna2 ,gisna3 ;contiene i numeri di snapshots C e D (snap2 - snap3)
giResAB ftgen 193,0,gispace, -2, 0 ;contiene il risultato del morphing A e B
giResCD ftgen 194,0,gispace, -2,0 ;contiene il risultato del morphing C e D
giVect ftgen 192,0,2,-2,giResAB,giResCD ;contiene le funzioni giResAB e giResCD
giResVect ftgen 195,0,gispace,-2,0 ;contiene l'interpolazione Y
```



Vectorial.csd

```
instr 9 ;Vectorial
ftmorf gkX, giSnapAB, giResAB
ftmorf gkX, giSnapCD, giResCD
ftmorf gkY, giVect, giResVect
vtabi 0,giResVect,iv1,iv2,iv3
FLsetVal_i iv1,gih1
FLsetVal_i iv2,gih2
FLsetVal_i iv3,gih3
endin
instr 10
tabw_i gisna0+i(gksnap0),0,giSnapAB
tabw_i gisna0+i(gksnap1),1,giSnapAB
tabw_i gisna0+i(gksnap2),0,giSnapCD
tabw_i gisna0+i(gksnap3),1,giSnapCD
endin
```



```
ktrig changed gkX,gkY
schedwhen ktrig, 0, 1, 9, 0, 1/giUpsliders ;call Hyper Vectorial instrument
```

Utilizzo delle MACRO_1 Draw_2.csd

```
giPixel init 80
;parametri utente
iDx init 10 ;posizione X oggetto
iDy init 10 ;posizione Y oggetto
iDw init 5 ;dimensione
iDh init iDw*giPixel*.5
#define MODULE(a'b'c'd'e'f'g'h'i'l'm'n'o'p'q'r's') #
FLcolor 210,210,210
knull1,gihband$b FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx, iDy
knull2,gihband$c FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*1), iDy
knull3,gihband$d FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*2), iDy
knull4,gihband$e FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*3), iDy
knull5,gihband$f FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*4), iDy
knull6,gihband$g FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*5), iDy
knull7,gihband$h FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*6), iDy
knull8,gihband$i FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*7), iDy
knull9,gihband$l FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*8), iDy
knull10,gihband$m FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*9), iDy
knull11,gihband$n FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*10), iDy
knull12,gihband$o FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*11), iDy
knull13,gihband$p FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*12), iDy
knull14,gihband$q FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*13), iDy
knull15,gihband$r FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*14), iDy
knull16,gihband$s FLslider " ", -1,1, 0, 4, -1, iDw, iDh, (${a*16*iDw)+iDx+(iDw*15), iDy
FLcolor -1
FLsetBox 1,gihband$b
FLsetBox 1,gihband$c
FLsetBox 1,gihband$d
FLsetBox 1,gihband$e
FLsetBox 1,gihband$f
FLsetBox 1,gihband$g
FLsetBox 1,gihband$h
FLsetBox 1,gihband$i
FLsetBox 1,gihband$l
FLsetBox 1,gihband$m
FLsetBox 1,gihband$n
FLsetBox 1,gihband$o
FLsetBox 1,gihband$p
FLsetBox 1,gihband$q
FLsetBox 1,gihband$r
FLsetBox 1,gihband$s
#

ginull_1 FLbox " ",8,gifont,11,iDw*giPixel+1,iDh+2,iDx-1,iDy-1
FLsetColor 210,210,210,ginull_1
FLsetAlign 6,ginull_1
$MODULE(0'1'2'3'4'5'6'7'8'9'10'11'12'13'14'15'16')
$MODULE(1'17'18'19'20'21'22'23'24'25'26'27'28'29'30'31'32')
$MODULE(2'33'34'35'36'37'38'39'40'41'42'43'44'45'46'47'48')
$MODULE(3'49'50'51'52'53'54'55'56'57'58'59'60'61'62'63'64')
$MODULE(4'65'66'67'68'69'70'71'72'73'74'75'76'77'78'79'80')
;$MODULE(5'81'82'83'84'85'86'87'88'89'90'91'92'93'94'95'96')
ginull_2 FLbox " ",8,gifont,11,iDw*giPixel,1,iDx,iDy+(iDh*.5)
ginull_3 FLbox " ",8,gifont,11,1,iDh,iDx+(iDw*giPixel*.25),iDy
ginull_4 FLbox " ",8,gifont,11,1,iDh,iDx+(iDw*giPixel*.5),iDy
ginull_5 FLbox " ",8,gifont,11,1,iDh,iDx+(iDw*giPixel*.75),iDy
```

Macro define
(fuori da FLpanel)

GUI generatio
(dentro FLpanel)

Utilizzo delle MACRO_2 Draw_2.csd

```
#define DRAW(a'b'c'd'e'f'g'h'i'l'm'n'o'p'q'r') #
if incr = $a igoto $a_0
if incr = $b igoto $b_1
if incr = $c igoto $c_2
if incr = $d igoto $d_3
if incr = $e igoto $e_4
if incr = $f igoto $f_5
if incr = $g igoto $g_6
if incr = $h igoto $h_7
if incr = $i igoto $i_8
if incr = $l igoto $l_9
if incr = $m igoto $m_10
if incr = $n igoto $n_11
if incr = $o igoto $o_12
if incr = $p igoto $p_13
if incr = $q igoto $q_14
if incr = $r igoto $r_15
igoto newblok$a
$a_0:
FLsetVal_i ival,gihband$a
igoto contin
$b_1:
FLsetVal_i ival,gihband$b
igoto contin
$c_2:
FLsetVal_i ival,gihband$c
igoto contin
$d_3:
=====
contin:
#
```

```
instr 500
imin init 1/kr
incr init -1
iWinFunc init i(gkWin)
imetà init ((giPixel-1)/kr)
irise init i(gkattack)*imetà
idec init i(gkdecay)*imetà
isus init imetà-(irise+idec)
kinchr linseg 0,irise,.5,isus,.5,idec,1
reset:
incr = incr+1
iWin tab_i i(kinchr),iWinFunc,1
ival init -(iWin)
;one drawing = 16 pixel
$DRAW(1'2'3'4'5'6'7'8'9'10'11'12'13'14'15'16')
$DRAW(17'18'19'20'21'22'23'24'25'26'27'28'29'30'31'32')
$DRAW(33'34'35'36'37'38'39'40'41'42'43'44'45'46'47'48')
$DRAW(49'50'51'52'53'54'55'56'57'58'59'60'61'62'63'64')
$DRAW(65'66'67'68'69'70'71'72'73'74'75'76'77'78'79'80')
contin:
if incr < giPixel kgoto salta
reinit stop
stop:
turnoff
salta:
reinit reset
endin

instr 1
ktrDraw changed gkattack,gkdecay,gkWin
schedk ktrDraw, +105,500, 0,.5 ;calling drawing instrument
endin
```

Numero funzione prototipo

Calcola le pendenze

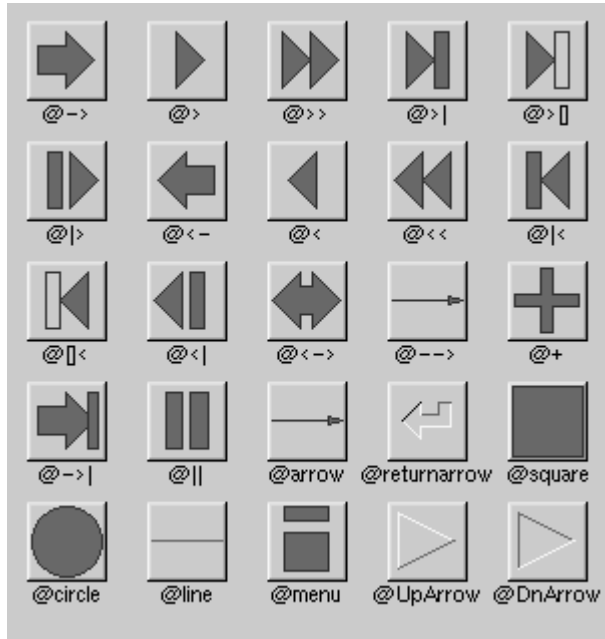
Phase della tabella

Ogni giro incrementa di 1

Legge i campioni

Verifica l'indice

Caratteri e simboli speciali



I simboli si visualizzano mettendo la @ come primo carattere della stringa; è possibile formattare il simbolo inserendo i valori tra la @ e il simbolo

Es:

"@+52>"

Size del simbolo (+ o -)

Rotazione di 45°

Simbolo

Attivazione symbol label

Da +[1-9] oppure -[1-9] il primo parametro riscalda il simbolo.

Da [1-9] ruota di 45 gradi il simbolo.

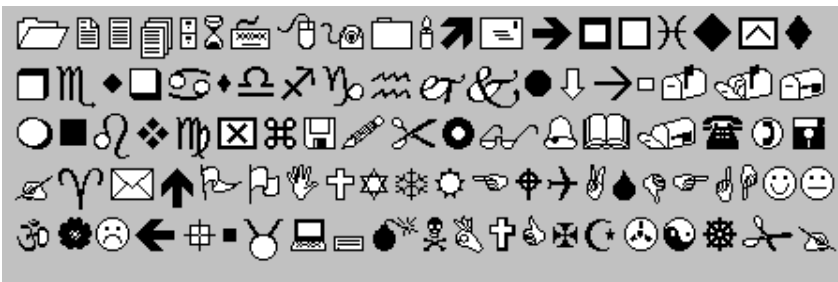


FontType.csd

Valori utili per ifont:

- 1 - helvetica (arial)
- 2 - helvetica bold
- 3 - helvetica italic
- 4 - helvetica bold italic
- 5 - courier
- 6 - courier bold
- 7 - courier italic
- 8 - courier bold italic
- 9 - times
- 10 - times bold
- 11 - times italic
- 12 - times bold italic
- 13 - symbol
- 14 - screen
- 15 - screen bold
- 16 - dingbats

Windings (font tipo 16-dingbats)



N.B. Non sono permessi i seguenti caratteri: \$ @ “ ;

Bugs ed espedienti

- Il parametro "*label*" di tutti i widgets, tra le virgolette va sempre inserito un carattere, per esempio uno spazio, non è permesso usare altre virgolette.
 - `FLbutton` del *type* 2 e `FLbutBank` non possono essere *checked* da `FLsetVal_i` ma possono essere *unchecked*
 - **N.B.** `FLbutton` *type* 2 con *iopcode* posto a +105 oppure 0, il widget chiama lo strumento quando posto in *checked*, ma lo chiama anche quando *unchecked* (vedi `turnoffk`)
 - **N.B.** I valori *imin* e *imax* di tutti i widgets devono rispettare l'ordine di grandezza, se diverso le inizializzazioni mediante `FLsetVal` non funzionano.
 - Se in `FLslider` uso una curvatura esponenziale (vedi *iexp*) il valore minimo non può essere zero ma tendere a zero es: 0.0000001.
 - **N.B.** Le tipologie di bottoni possono essere: 1 *normal button*, 2 *light button*, 3 *check button* e 4 *round button*, Aggiungendo 10 all'argomento *itype* (es: settando 11 per il tipo 1, 12 per il 2, 13 per il 3 ecc..) è possibile saltare il salvataggio del *widget* quando si usano *getting/setting* degli *snapshots* opcodes, (vedi `FLsavesnap`).
 - In `FLsidBnk` *ityetable* può essere solo settato a 1, altri valori provocano delle alterazioni grafiche.
-



CSOUND AV