```
<CsoundSynthesizer>
<CsOptions>
-m128
;--realtime ;audio RT priority as option if performance is not sufficient
</CsOptions>
<CsInstruments>

sr = 44100
ksmps = 128
nchnls = 8
0dbfs = 1

/*****************************************************************************/
/*****************************************************************************/
/*****************************************************************************/
/*                                                                           */
/*                      GUENTER STEINKE "ARCADE" (1991)                      */
/*                                                                           */
/*               Version for Csound by Joachim Heintz (2019)                 */
/*                                                                           */
/*    with Marijana Janevska, Daria Cheikh-Sarraf, Farhad Ilaghi Hosseini    */
/*         Philipp Henkel, Shadi Kassaee, Dilxat Dawut, Hunjoo Jung          */
/*         (members of a seminar at HMTM Hannover in winter 2018/19)         */
/*                                                                           */
/*      Requirements: Csound 6.14 or higher and CsoundQt 0.9.8 or higher     */
/*                                                                           */
/*****************************************************************************/
/*****************************************************************************/



/*****************************************************************************/
/*****************************************************************************/
/*                                                                           */
/*                                I/O SETTINGS                               */

;software and hardware outputs
;only valid for alternative output selection
;see instrument Output around line 2850
gi_OutChn_1 = 1
gi_OutChn_2 = 2
gi_OutChn_3 = 3
gi_OutChn_4 = 4
gi_OutChn_5 = 5
gi_OutChn_6 = 6
gi_OutChn_7 = 7
gi_OutChn_8 = 8

;different output settings for 4, 6 or 8 speakers
vbaplsinit 2.01, 4, -45, 45, 135, -135
vbaplsinit 2.02, 6, -30, 30, 90, 150, -150, -90
vbaplsinit 2.03, 8, -22, 23, 68, 113, 158, -157, -112, -67


/*                                                                           */
/*****************************************************************************/
/*****************************************************************************/



/*****************************************************************************/
/*****************************************************************************/
/*                                                                           */
/*                             OPCODE DEFINITIONS                            */

/* all matrix instruments */
gS_Mtxs[] fillarray "Mtx_1", "Mtx_2", "Mtx_3", "Mtx_4", "Mtx_5", "Mtx_6", "Mtx_7",
      "Mtx_8", "Mtx_9", "Mtx_10", "Mtx_11", "Mtx_12", "Mtx_13", "Mtx_14", "Mtx_15"
```

```
74
75 /* a very long time without using -1 */
76 #define L # 999999 #
77
78
79 opcode setTo, k, kii
80  kVar, iTarget, iTime xin
81  iStart = i(kVar)
82  kOut linseg iStart, iTime, iTarget
83  xout kOut
84 endop
85
86 opcode TurOffOtherMtxs, 0, S[]S
87 //turnoff all running instruments from S_all[] except S_this
88  S_all[], S_this xin
89  indx = 0
90  while indx < lenarray(S_all) do
91   if strcmp(S_this,S_all[indx])!= 0 then
92    if active:i(S_all[indx])>0 then
93     schedule -nstrnum(S_all[indx]), 0, 0
94     printf_i "  %s turned off\n", 1, S_all[indx]
95    endif
96   endif
97   indx += 1
98  od
99 endop
100
101 opcode TurnOnFilter, 0, i
102  iNumFilterBands xin
103  instnc = 1
104  while instnc < iNumFilterBands+1 do
105   schedule("Filt_A",0,$L,instnc)
106   schedule("Filt_B",0,$L,instnc)
107   instnc += 1
108  od
109 endop
110
111 opcode TurnOffFilter, 0, 0
112  turnoff2 "Filt_A", 0, 0
113  turnoff2 "Filt_B", 0, 0
114 endop
115
116 opcode SetFilterInput, 0, i
117  iNumFilterBands xin
118  indx = 0
119  while indx < iNumFilterBands do
120   chnset 0, sprintf("Filt_A_%d",indx+1)
121   chnset 0, sprintf("Filt_B_%d",indx+1)
122   indx += 1
123  od
124 endop
125
126 opcode TurnOnRev, 0, o
127  iWhen xin
128  schedule "Rev_AB", iWhen, -1
129 endop
130
131 opcode Report, 0, S
132  S_ins xin
133  kActive active S_ins
134  printf "  Running instances of %s: %d\n", changed(kActive), S_ins, kActive
135 endop
136
137 opcode CsQtMeter, 0, SSak
138  S_chan_sig, S_chan_over, aSig, kTrig  xin
139  iDbRange = 60 ;shows 60 dB
140  iHoldTim = 1 ;seconds to "hold the red light"
141  kOn init 0
142  kTim init 0
143  kStart init 0
144  kEnd init 0
145  kMax max_k aSig, kTrig, 1
146  if kTrig == 1 then
147   chnset (iDbRange + dbfsamp(kMax)) / iDbRange, S_chan_sig
```

```
  if kOn == 0 && kMax > 1 then
   kTim = 0
   kEnd = iHoldTim
   chnset k(1), S_chan_over
   kOn = 1
  endif
  if kOn == 1 && kTim > kEnd then
   chnset k(0), S_chan_over
   kOn =  0
  endif
 endif
 kTim += ksmps/sr
endop


/*                                                                              */
/********************************************************************************/
/********************************************************************************/


/********************************************************************************/
/********************************************************************************/
/*                                                                              */
/*                        GLOBAL VARIABLES AND SETTINGS                         */


/* Input */
ga_Mic init 0
gk_LiveAmp_reductDb init 0 ;used for fade out live amp in cue 93

/* Delay */
gi_Del_MaxDelTim init 15 ;sec
gi_Del_FeedbLev init 0.94 ;which feedback level for "on"
gk_Del1A_DelTim init 1
gk_Del1B_DelTim init 1
gk_Del2A_DelTim init 1
gk_Del2B_DelTim init 1
gk_Del1A_FbLev init 0
gk_Del1B_FbLev init 0
gk_Del2B_FbLev init 0
gk_Del2A_FbLev init 0
gk_Del1A_VolIn init 0
gk_Del1B_VolIn init 0
gk_Del2A_VolIn init 0
gk_Del2B_VolIn init 0
gk_Del1A_VolOut init 0
gk_Del1B_VolOut init 0
gk_Del2B_VolOut init 0
gk_Del2A_VolOut init 0
ga_Del1A_out init 0
ga_Del1B_out init 0
ga_Del2A_out init 0
ga_Del2B_out init 0
ga_Del1_out init 0 ;sum of 1A and 1B
ga_Del2_out init 0 ;2A+2B
ga_Del_out init 0 ;sum of all del output


/* Harmonizer */
gi_Harm_MaxDelTim init 10500 ;msec
gk_Harm1A_Pch init 1
gk_Harm1B_Pch init 1
gk_Harm2A_Pch init 1
gk_Harm2B_Pch init 1
gk_Harm1A_Del init 1
gk_Harm1B_Del init 1
gk_Harm2A_Del init 1
gk_Harm2B_Del init 1
gk_Harm1A_VolIn init 0
gk_Harm1B_VolIn init 0
gk_Harm2A_VolIn init 0
gk_Harm2B_VolIn init 0
```

```
gk_Harm1A_VolOut init 0
gk_Harm1B_VolOut init 0
gk_Harm2A_VolOut init 0
gk_Harm2B_VolOut init 0
ga_Harm_in init 0
ga_Harm1A_out init 0
ga_Harm1B_out init 0
ga_Harm2A_out init 0
ga_Harm2B_out init 0
ga_Harm1_out init 0
ga_Harm2_out init 0
ga_Harm_out init 0


/* Filter */
;filter programs A and B
giFilt_A ftgen 0, 0, 0, -23, "filter_A_47.csv"
giFilt_B ftgen 0, 0, 0, -23, "filter_B_47.csv"
;durations for filter programs 1-14 and 15-32 in sec
gk_Filt_Seq_1[] fillarray 8, 6.5, 4.5, 3, 5.5, 2.5, 2, 3, 9, 8, 7, 5, 9.5, 9.9
gk_Filt_Seq_2[] fillarray 9, 4, 6, 5.5, 4.5, 6, 8, 7, 7, 4, 6, 7, 2, 5, 7, 6, 3, 8
gi_Filt_NumBands = 47 ;band 48 can be ignored as always closed
gi_Filt_FadeTim = 5
gk_Filt_Q init 10
gk_Filt_VolOut init 0
ga_Filt_in init 0
ga_FiltA_out init 0
ga_FiltB_out init 0
ga_Filt_out init 0


/* Reverb */
gk_RevA_VolIn init 0
gk_RevB_VolIn init 0
gk_RevA_VolOut init 0
gk_RevB_VolOut init 0
ga_Rev_in init 0
ga_RevA_out init 0
ga_RevB_out init 0


/* Halafon */
gk_HalaA_Speed init 1/20 ;cycles per sec (very slow)
gk_HalaB_Speed init 1/5 ;medium
gk_HalaC_Speed init 1 ;fast
gk_HalaA_Dir init 1 ;1=clockwise -1=counterclock
gk_HalaB_Dir init -1
gk_HalaC_Dir init 1
gk_Hala_Spread init 0 ;usually 0, maximum 100
gk_HalaA_VolOut init 0
gk_HalaB_VolOut init 0
gk_HalaC_VolOut init 0
ga_HalaA_in init 0
ga_HalaB_in init 0
ga_HalaC_in init 0
ga_HalaA_out init 0
ga_HalaB_out init 0
ga_HalaC_out init 0


/* Internal 6 Channels Out */
ga_Chn1_in init 0
ga_Chn2_in init 0
ga_Chn3_in init 0
ga_Chn4_in init 0
ga_Chn5_in init 0
ga_Chn6_in init 0


/* fft settings */
giFFTsize = 1024
giFFThopsize = giFFTsize/4


```

```
296 /*                                                                            */
297 /****************************************************************************/
298 /****************************************************************************/
299
300
301
302
303 /****************************************************************************/
304 /****************************************************************************/
305 /*                                                                            */
306 /*                               SOFTWARE CHANNELS                            */
307
308 /* input and output */
309 chn_k "count", 3
310 chn_k "next_count", 3
311 chn_a "filt_A_collect", 3
312 chn_a "filt_B_collect", 3
313
314 /* output (= show signals in widgets) */
315 chn_k "Del_1A_in", 2
316 chn_k "Del_1B_in", 2
317 chn_k "Del_2A_in", 2
318 chn_k "Del_2B_in", 2
319 chn_k "Del_1A_out", 2
320 chn_k "Del_1B_out", 2
321 chn_k "Del_2A_out", 2
322 chn_k "Del_2B_out", 2
323 chn_k "Harm_1A_in", 2
324 chn_k "Harm_1B_in", 2
325 chn_k "Harm_2A_in", 2
326 chn_k "Harm_2B_in", 2
327 chn_k "Harm_1A_out", 2
328 chn_k "Harm_1B_out", 2
329 chn_k "Harm_2A_out", 2
330 chn_k "Harm_2B_out", 2
331 chn_k "Filt_out", 2
332 chn_k "Rev_A_in", 2
333 chn_k "Rev_B_in", 2
334 chn_k "Rev_A_out", 2
335 chn_k "Rev_B_out", 2
336 chn_k "Hala_A", 2
337 chn_k "Hala_B", 2
338 chn_k "Hala_C", 2
339 chn_k "micA_in_disp", 2
340 chn_k "micA_in_over_disp", 2
341 chn_k "micB_in_disp", 2
342 chn_k "micB_in_over_disp", 2
343 chn_k "micSum_in_disp", 2
344 chn_k "micSum_in_over_disp", 2
345 chn_k "Del_1_disp", 2
346 chn_k "Del_1_over_disp", 2
347 chn_k "Del_2_disp", 2
348 chn_k "Del_2_over_disp", 2
349 chn_k "Harm_1_disp", 2
350 chn_k "Harm_1_over_disp", 2
351 chn_k "Harm_2_disp", 2
352 chn_k "Harm_2_over_disp", 2
353 chn_k "Rev_disp", 2
354 chn_k "Rev_over_disp", 2
355 chn_k "Filt_disp", 2
356 chn_k "Filt_over_disp", 2
357 chn_k "Hala_A_disp", 2
358 chn_k "Hala_A_over_disp", 2
359 chn_k "Hala_B_disp", 2
360 chn_k "Hala_B_over_disp", 2
361 chn_k "Hala_C_disp", 2
362 chn_k "Hala_C_over_disp", 2
363 chn_k "HalaA_speed_disp", 2
364 chn_k "HalaB_speed_disp", 2
365 chn_k "HalaC_speed_disp", 2
366 chn_k "out_1_disp", 2
367 chn_k "out_1_over_disp", 2
368 chn_k "out_2_disp", 2
369 chn_k "out_2_over_disp", 2
```

```
chn_k "out_3_disp", 2
chn_k "out_3_over_disp", 2
chn_k "out_4_disp", 2
chn_k "out_4_over_disp", 2
chn_k "out_5_disp", 2
chn_k "out_5_over_disp", 2
chn_k "out_6_disp", 2
chn_k "out_6_over_disp", 2
chn_k "out_7_disp", 2
chn_k "out_7_over_disp", 2
chn_k "out_8_disp", 2
chn_k "out_8_over_disp", 2
chn_k "show_mtx", 2
chn_S "Playback_time", 2
chn_k "record", 2
chn_k "playback", 2
chn_S "message", 2

;/* input (= get values from widgets (mostly via midi)) */
chn_k "Del_out_midi", 1
chn_k "Harm_out_midi", 1
chn_k "Filt_out_midi", 1
chn_k "Rev_out_midi", 1
chn_k "Hala_A_out_midi", 1
chn_k "Hala_B_out_midi", 1
chn_k "Hala_C_out_midi", 1
chn_k "Sum_out_midi", 1
chn_k "Rev_reduct_midi", 1
chn_k "Hala_A_speed_midi", 1
chn_k "Hala_B_speed_midi", 1
chn_k "Hala_C_spedd_midi", 1
chn_k "micA_gain_midi", 1
chn_k "micB_gain_midi", 1
chn_k "live_amp_midi", 1
chn_k "Playback_Vol_Db", 1
chn_k "micA_input_chn", 1
chn_k "micB_input_chn", 1
chn_k "mic_select", 1
chn_k "live_amp_select", 1
chn_k "filt_q", 1
chn_k "out_setup", 1


/*                                                                              */
/********************************************************************************/
/********************************************************************************/



/********************************************************************************/
/********************************************************************************/
/*                                                                              */
/*                                    INIT                                      */

/*** ALWAYS ON INSTRUMENTS ***/
alwayson "Trigger_Cues"
alwayson "Midi"
alwayson "Input"
alwayson "Del_1A"
alwayson "Del_1B"
alwayson "Del_2A"
alwayson "Del_2B"
alwayson "Del_sums"
alwayson "Harm_1A"
alwayson "Harm_1B"
alwayson "Harm_2A"
alwayson "Harm_2B"
alwayson "Harm_sums"
alwayson "Filt"
schedule("Chn_1",.1,-1)
schedule("Chn_2",.1,-1)
schedule("Chn_3",.1,-1)
```

```
444 schedule("Chn_4",.1,-1)
445 schedule("Chn_5",.1,-1)
446 schedule("Chn_6",.1,-1)
447 schedule("Output",.1,-1)
448 alwayson "Show"
449
450 /*** PREALLOCATION ***/
451 prealloc "Filt_A", 47
452 prealloc "Filt_B", 47
453 prealloc "Hala_A", 1
454 prealloc "Hala_B", 1
455 prealloc "Hala_C", 1
456 prealloc "Rev_AB", 1
457
458
459
460 /*** INIT ***/
461 instr Init
462
463  //initialize some widgets
464  chnset 1, "next_count"
465  chnset 0, "count"
466  chnset 0, "record"
467  chnset 0, "playback"
468
469  //initialize globale output array
470  gi_Setup chnget "out_setup"
471  gi_Setup += 1
472  iNumChnls = gi_Setup*2 + 2
473  ga_Out[] init iNumChnls
474  ga_Clear[] init iNumChnls
475  chnset sprintf("Output setup = %d\nRendering %d output channels.",
476                 gi_Setup,iNumChnls), "message"
477
478  turnoff
479
480 endin
481 schedule "Init", 0, 1
482
483
484 /*                                                                          */
485 /**************************************************************************/
486 /**************************************************************************/
487
488
489
490 /**************************************************************************/
491 /**************************************************************************/
492 /*                                                                          */
493 /*                      PRERECORDED VERSION FOR TESTING                     */
494
495
496 instr PlaybackRec
497
498  turnoff2 "Input", 0, 0
499
500  S_recording = "Steinke_Arcade_Cello_only.wav"
501 iCueTimes[] fillarray 0.0, 0.72, 4.99, 9.63, 15.19, 20.13, 26.25, 32.55, 35.37, 37.84,
502    42.99, 45.07, 47.32, 50.99, 53.44, 58.39, 59.78, 62.04, 72.88, 81.6,
503    85.26, 101.1, 114.43, 116.08, 125.64, 127.9, 131.24, 135.07, 138.12, 142.0,
504    146.56, 147.09, 150.82, 153.68, 158.55, 173.32, 174.79, 177.96, 180.38, 190.63,
505    213.46, 217.93, 221.31, 227.06, 230.21, 238.53, 240.1, 245.75, 249.66, 257.71,
506    259.87, 270.06, 272.94, 280.04, 284.84, 293.2, 299.73, 356.47, 374.46, 375.14,
507    378.16, 381.03, 383.76, 388.16, 391.86, 394.03, 395.8, 399.25, 404.19, 409.26,
508    410.87, 414.79, 416.25, 420.33, 421.47, 424.07, 428.02, 433.68, 438.03, 441.43,
509    442.87, 446.15, 457.04, 462.08, 467.32, 478.59, 489.28, 493.9, 497.97, 504.82,
510    507.08, 510.77, 515.87, 544.17, 568.73, 575.92, 598.79, 617.02, 999
511
512  iStartCue chnget "next_count"
513  iTimeOffset = iCueTimes[iStartCue-1]
514  ga_Mic diskin S_recording, 1, iTimeOffset
515  kOutVolDb chnget "Playback_Vol_Db"
516  out ga_Mic*ampdb(kOutVolDb), ga_Mic*ampdb(kOutVolDb)
517
```

```
 kTime = timeinsts:k() + iTimeOffset
 kCue chnget "next_count"
 kNextTime = iCueTimes[kCue-1]

 if kTime > kNextTime then
  event "i", sprintfk("Cue_%d",kCue), 0, .1
  chnset kCue, "count"
  chnset kCue+1, "next_count"
 endif

 S_time sprintfk "%02d : %02d", int(kTime/60), int(kTime%60)
 chnset S_time, "Playback_time"

endin

/*                                                                            */
/******************************************************************************/
/******************************************************************************/



/******************************************************************************/
/******************************************************************************/
/*                                                                            */
/*                                    CUES                                    */


instr Trigger_Cues

 kKey sensekey

 if kKey > 0 then

  kCounter chnget "next_count"

  //plus key: counter+1 (without action)
  if kKey == 43 then
   kCounter += 1
   chnset kCounter, "next_count"

  //minus key: counter-1
  elseif kKey == 45 then
   kCounter -= 1
   chnset kCounter, "next_count"

  //space bar: trigger events
  elseif kKey == 32 then

   event "i", sprintfk("Cue_%d",kCounter), 0, .1
   kCounter += 1
   chnset kCounter, "next_count"
   chnset kCounter-1, "count"

  endif

 endif

endin


instr Cue_0
 //dummy
endin

instr Cue_1

 puts "Cue_1", 1
 schedule "Mtx_1", 0, -1

 //delay
 gk_Del1A_DelTim = 7
 gk_Del1B_DelTim = 8.3
 gk_Del2A_DelTim = 10.7
```

```
 gk_Del2B_DelTim = 12
 gk_Del1A_FbLev = 0
 gk_Del1B_FbLev = 0
 gk_Del2B_FbLev = 0
 gk_Del2A_FbLev = 0
 gk_Del1A_VolIn = 0
 gk_Del1B_VolIn = 0
 gk_Del2A_VolIn = 0
 gk_Del2B_VolIn = 0
 gk_Del1A_VolOut = 1
 gk_Del1B_VolOut = 1
 gk_Del2A_VolOut = 1
 gk_Del2B_VolOut = 1

 //harmonizer
 gk_Harm1A_Pch = 1.015
 gk_Harm1B_Pch = 0.985
 gk_Harm2A_Pch = 1.031
 gk_Harm2B_Pch = 0.971
 gk_Harm1A_Del = 200
 gk_Harm1B_Del = 350
 gk_Harm2A_Del = 580
 gk_Harm2B_Del = 800
 gk_Harm1A_VolIn = 1
 gk_Harm1B_VolIn = 1
 gk_Harm2A_VolIn = 1
 gk_Harm2B_VolIn = 1
 gk_Harm1A_VolOut = 1
 gk_Harm1B_VolOut = 1
 gk_Harm2A_VolOut = 1
 gk_Harm2B_VolOut = 1

 turnoff

endin

instr Cue_2
 puts "Cue_2", 1
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
endin

instr Cue_3
 puts "Cue_3", 1
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
endin

instr Cue_4
 puts "Cue_4", 1
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
endin

instr Cue_5
 puts "Cue_5", 1
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 schedule "cue_5a", 3, 1
endin

instr cue_5a
 puts "  Cue_5a", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
endin

instr Cue_6
 puts "Cue_6", 1
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
endin

instr Cue_7
 puts "Cue_7", 1
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
```

```
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
 schedule "Cue_7a", 2, 1
endin

instr Cue_7a
 puts "  Cue_7a", 1
 schedule "Mtx_2", 0, -1
 //harmonizer prog 2
 gk_Harm1A_Pch = 1.031
 gk_Harm1B_Pch = 0.958
 gk_Harm2A_Pch = 1.066
 gk_Harm2B_Pch = 0.946
 turnoff
endin

instr Cue_8
 puts "Cue_8", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
endin

instr Cue_9
 puts "Cue_9", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_10
 puts "Cue_10", 1
 schedule "Mtx_3", 0, -1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
 //harmonizer prog 3
 gk_Harm1A_Pch = 1.066
 gk_Harm1B_Pch = 0.971
 gk_Harm2A_Pch = 1.048
 gk_Harm2B_Pch = 0.925
endin

instr Cue_11
 puts "Cue_11", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
endin

instr Cue_12
 puts "Cue_12", 1
 schedule "Mtx_4", 0, -1
 //harmonizer prog 4
 gk_Harm1A_Pch = 1.048
 gk_Harm1B_Pch = 0.946
 gk_Harm2A_Pch = 1.085
 gk_Harm2B_Pch = 0.935
 turnoff
endin

instr Cue_13
 puts "Cue_13", 1
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
endin

instr Cue_14
 puts "Cue_14", 1
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
endin

instr Cue_15
 puts "Cue_15", 1
```

```
 schedule "Mtx_5", 0, -1
 //harmonizer prog 5
 gk_Harm1A_Pch = 1.105
 gk_Harm1B_Pch = 0.925
 gk_Harm2A_Pch = 1.015
 gk_Harm2B_Pch = 0.985
 turnoff
endin

instr Cue_16
 puts "Cue_16", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
endin

instr Cue_17
 puts "Cue_17", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
endin

instr Cue_18
 puts "Cue_18", 1
 schedule "Mtx_6", 0, -1
 //harmonizer prog 6
 gk_Harm1A_Pch = 1.085
 gk_Harm1B_Pch = 0.935
 gk_Harm2A_Pch = 1.105
 gk_Harm2B_Pch = 0.958
 turnoff
endin

instr Cue_19
 puts "Cue_19", 1
 schedule "Mtx_7", 0, -1
 //del prog 2
 gk_Del1A_DelTim = 3.5
 gk_Del1B_DelTim = 4.5
 gk_Del2A_DelTim = 8.5
 gk_Del2B_DelTim = 10.5
 //harmonizer prog 7
 gk_Harm1A_Pch = 1.025
 gk_Harm1B_Pch = 0.951
 gk_Harm2A_Pch = 1.105
 gk_Harm2B_Pch = 0.958
 gk_Harm1A_Del = 1 ;really
 gk_Harm1B_Del = 3500
 //open del 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
endin

instr Cue_20
 puts "Cue_20", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_21
 puts "Cue_21", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
 //harmonizer prog 8
 gk_Harm1A_Pch = 1.025
 gk_Harm1B_Pch = 0.951
 gk_Harm2A_Pch = 1.079
 gk_Harm2B_Pch = 0.906
 gk_Harm2A_Del = 6500
```

```
 gk_Harm2B_Del = 10500
endin


instr Cue_22
 puts "Cue_22", 1
 gk_Del2A_FbLev = setTo(gk_Del2A_FbLev,gi_Del_FeedbLev,p3)
 gk_Del2B_FbLev = setTo(gk_Del2B_FbLev,gi_Del_FeedbLev,p3)
endin

instr Cue_23
 puts "Cue_23", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
endin

instr Cue_24
 puts "Cue_24", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
endin

instr Cue_25
 puts "Cue_25", 1
 //close all Del in
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
 //set Harm Prog 9
 gk_Harm1A_Pch = 1.167
 gk_Harm1B_Pch = 0.928
 gk_Harm2A_Pch = 1.051
 gk_Harm2B_Pch = 0.828
 gk_Harm1A_Del = 125
 gk_Harm1B_Del = 125
 gk_Harm2A_Del = 125
 gk_Harm2B_Del = 125
 //set Mtx_8
 schedule "Mtx_8", p3, -1
endin

instr Cue_26
 puts "Cue_26", 1
 //set Del 1 Prog 3
 gk_Del1A_DelTim = 7.5
 gk_Del1B_DelTim = 6
 //set Harm Prog 10
 gk_Harm1A_Pch = 1.079
 gk_Harm1B_Pch = 0.885
 gk_Harm2A_Pch = 1.137
 gk_Harm2B_Pch = 0.906
 gk_Harm1A_Del = 180
 gk_Harm1B_Del = 220
 gk_Harm2A_Del = 265
 gk_Harm2B_Del = 300
 //start filter (no input yet)
 TurnOnFilter(gi_Filt_NumBands)
 turnoff
endin

instr Cue_27
 puts "Cue_27", 1
 //set Harm Prog 11
 gk_Harm1A_Pch = 1.230
 gk_Harm1B_Pch = 0.846
 gk_Harm2A_Pch = 1.108
 gk_Harm2B_Pch = 0.865
 gk_Harm1A_Del = 200
 gk_Harm1B_Del = 330
```

```
 gk_Harm2A_Del = 480
 gk_Harm2B_Del = 600
 //set filter to initial values (all bands open)
 SetFilterInput(gi_Filt_NumBands)
 //start all Halafons (A and B get signal in Mtx_9)
 schedule "Hala_A", 0, -1
 schedule "Hala_B", 0, -1
 schedule "Hala_C", 0, -1
 turnoff
endin

instr Cue_28
 puts "Cue_28", 1
 //set Mtx_9
 schedule "Mtx_9", 0, -1
 //open filter output
 gk_Filt_VolOut = setTo(gk_Filt_VolOut,1,p3)
 //start Filtersequence
 schedule "Filt_Seq_1", 0, -1
 //open Hala A and B output
 gk_HalaA_VolOut = setTo(gk_HalaA_VolOut,1,p3)
 gk_HalaB_VolOut = setTo(gk_HalaB_VolOut,1,p3)
 //open Del_1A_in and close after 2 sec
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 schedule "Cue_28a", 2, 1
endin

instr Cue_28a
 puts "  Cue_28a", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
endin

instr Cue_29
 puts "Cue_29", 1
 //open Del_1B_in and close after 1 sec
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 schedule "Cue_29a", 1, 1
endin

instr Cue_29a
 puts "  Cue_29a", 1
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_30
 puts "Cue_30", 1
 //open Del_1A_in and close after 2 sec
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 schedule "Cue_30a", 2, 1
endin

instr Cue_30a
 puts "  Cue_30a", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
endin

instr Cue_31
 puts "Cue_31", 1
 //open Del_1B_in
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
endin

instr Cue_32
 puts "Cue_32", 1
 //close Del_1B_in
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
 //open Del_1A_in and close after3 sec
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 schedule "Cue_32a", 3, 1
endin

instr Cue_32a
 puts "  Cue_32a", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
```

```
endin

instr Cue_33
 puts "Cue_33", 1
 //open Del_1B_in and close after 1 sec
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 schedule "Cue_33a", 1, 1
endin

instr Cue_33a
 puts "  Cue_33a", 1
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_34
 puts "Cue_34", 1
 //open Del_1A_in
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
endin

instr Cue_35
 puts "Cue_35", 1
 //close Del_1A_in
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
endin

instr Cue_36
 puts "Cue_36", 1
 //open Del_1B_in
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 //set Harm Prog 12
 gk_Harm1A_Pch = 1.198
 gk_Harm1B_Pch = 0.828
 gk_Harm2A_Pch = 0.951
 gk_Harm2B_Pch = 1.167
 gk_Harm1A_Del = 150
 gk_Harm1B_Del = 220
 gk_Harm2A_Del = 300
 gk_Harm2B_Del = 400
endin

instr Cue_37
 puts "Cue_37", 1
 //close Del_1B_in
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_38
 puts "Cue_38", 1
 //open Del_1B_in
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
endin

instr Cue_39
 puts "Cue_39", 1
 //close Del_1B_in
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_40
 puts "Cue_40", 1
 //set Mtx_10
 schedule "Mtx_10", 0, -1
 //set Del 1 Prog 4
 gk_Del1A_DelTim = 6.3
 gk_Del1B_DelTim = 8
 turnoff
endin

instr Cue_41
 puts "Cue_41", 1
 //mute filter output
 gk_Filt_VolOut = setTo(gk_Filt_VolOut,0,p3)
 //stop feedback (del 2)
```

```
 gk_Del2A_FbLev = setTo(gk_Del2A_FbLev,0,p3)
 gk_Del2B_FbLev = setTo(gk_Del2B_FbLev,0,p3)
 //close del 2 in and out
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
 gk_Del2A_VolOut = setTo(gk_Del2A_VolOut,0,p3)
 gk_Del2B_VolOut = setTo(gk_Del2B_VolOut,0,p3)
 //set Harm Prog 13
 gk_Harm1A_Pch = 1.105
 gk_Harm1B_Pch = 0.935
 gk_Harm2A_Pch = 1.031
 gk_Harm2B_Pch = 0.971
 gk_Harm1A_Del = 400
 gk_Harm1B_Del = 620
 gk_Harm2A_Del = 830
 gk_Harm2B_Del = 1200
 //set Mtx_11 (Hala get no more signal)
 schedule "Mtx_11", p3, -1
 //close Hala A and B
 gk_HalaA_VolOut = setTo(gk_HalaA_VolOut,0,p3)
 gk_HalaB_VolOut = setTo(gk_HalaB_VolOut,0,p3)
 //open Del_1 in (A and B)
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
endin


instr Cue_42
 puts "Cue_42", 1
 //close Del_1 in
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
 //set Del 2 Prog 3
 gk_Del2A_DelTim = 10.3
 gk_Del2B_DelTim = 13
 //turn off filter
 schedule("Cue_42a",0,1)
endin

instr Cue_42a
 TurnOffFilter
 turnoff
endin

instr Cue_43
 puts "Cue_43", 1
 //open Del_1 in and close after 2 sec
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 schedule "Cue_43a", 2, 1
endin

instr Cue_43a
 puts "  Cue_43a", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_44
 puts "Cue_44", 1
 //open Del_2 in
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
endin

instr Cue_45
 puts "Cue_45", 1
 //closes Del_2 in
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
 //fade in Del_2 out
 schedule("Cue_45a", 0, 5)
 //open Del_1 in
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
```

```
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 //set Harm Prog 14
 gk_Harm1A_Pch = 1.066
 gk_Harm1B_Pch = 0.958
 gk_Harm2A_Pch = 1.048
 gk_Harm2B_Pch = 0.946
 gk_Harm1A_Del = 300
 gk_Harm1B_Del = 500
 gk_Harm2A_Del = 630
 gk_Harm2B_Del = 800
endin

instr Cue_45a
 puts "  Cue_45a", 1
 //fade in del 2 out
 gk_Del2A_VolOut linseg 0, p3, 1
 gk_Del2B_VolOut linseg 0, p3, 1
endin

instr Cue_46
 puts "Cue_46", 1
 //close del 1 in
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_47
 puts "Cue_47", 1
 //open all del in
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
endin

instr Cue_48
 puts "Cue_48", 1
 //set Harm Prog 15
 gk_Harm1A_Pch = 1.048
 gk_Harm1B_Pch = 0.971
 gk_Harm2A_Pch = 1.015
 gk_Harm2B_Pch = 0.985
 gk_Harm1A_Del = 200
 gk_Harm1B_Del = 350
 gk_Harm2A_Del = 480
 gk_Harm2B_Del = 600
 //close Del_2B in after 1 sec
 schedule "Cue_48a", 1, 1
 turnoff
endin

instr Cue_48a
 puts "  Cue_48a", 1
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
endin

instr Cue_49
 puts "Cue_49", 1
 //close Del_2A_in
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
 //closes Del_1B_in after 3.5 sec
 schedule "Cue_49a", 3.5, 1
 //close Del_1A_in after 5 sec
 schedule "Cue_49b", 5, 1
endin

instr Cue_49a
 puts "  Cue_49a", 1
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,0,p3)
endin

instr Cue_49b
 puts "  Cue_49b", 1
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
```

```
1184 endin
1185
1186 instr Cue_50
1187  puts "Cue_50", 1
1188  //set Mtx 12
1189  schedule("Mtx_12",0,-1)
1190  //open Hala B and C, close A
1191  gk_HalaB_VolOut = setTo(gk_HalaB_VolOut,1,p3)
1192  gk_HalaC_VolOut = setTo(gk_HalaC_VolOut,1,p3)
1193  gk_HalaA_VolOut = setTo(gk_HalaA_VolOut,0,p3)
1194 endin
1195
1196 instr Cue_51
1197  puts "Cue_51", 1
1198  //set Harm Prog 16
1199  gk_Harm1A_Pch = 0.885
1200  gk_Harm1B_Pch = 1.230
1201  gk_Harm2A_Pch = 0.951
1202  gk_Harm2B_Pch = 1.108
1203  gk_Harm1A_Del = 200
1204  gk_Harm1B_Del = 350
1205  gk_Harm2A_Del = 480
1206  gk_Harm2B_Del = 600
1207  //open Del_2A_in
1208  gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
1209 endin
1210
1211 instr Cue_52
1212  puts "Cue_52", 1
1213  //close Del_2A_in
1214  gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
1215 endin
1216
1217 instr Cue_53
1218  puts "Cue_53", 1
1219  //open Del_2B and clos after 4 sec
1220  gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
1221  schedule("Cue_53a",4,1)
1222 endin
1223
1224 instr Cue_53a
1225  puts "  Cue_53a", 1
1226  gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
1227 endin
1228
1229 instr Cue_54
1230  puts "Cue_54", 1
1231  //open Del_1A
1232  gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
1233  //set Harm Prog 17 after 1.5 sec
1234  schedule("Cue_54a",1.5,1)
1235 endin
1236
1237 instr Cue_54a
1238  puts "  Cue_54a", 1
1239  //set Harm Prog 17
1240  gk_Harm1A_Pch = 0.810
1241  gk_Harm1B_Pch = 0.906
1242  gk_Harm2A_Pch = 1.263
1243  gk_Harm2B_Pch = 1.079
1244  gk_Harm1A_Del = 175
1245  gk_Harm1B_Del = 290
1246  gk_Harm2A_Del = 410
1247  gk_Harm2B_Del = 550
1248  turnoff
1249 endin
1250
1251 instr Cue_55
1252  puts "Cue_55", 1
1253  //close Del_1A_in
1254  gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,0,p3)
1255 endin
1256
1257 instr Cue_56
```

```
 puts "Cue_56", 1
 p3 = 5
 //set Del 1 Prog 5 with soft transition
 gk_Del1A_DelTim = setTo(gk_Del1A_DelTim,9,p3)
 gk_Del1B_DelTim = setTo(gk_Del1B_DelTim,15,p3)
 //set Harm Prog 18
 gk_Harm1A_Pch = 0.928
 gk_Harm1B_Pch = 0.793
 gk_Harm2A_Pch = 1.198
 gk_Harm2B_Pch = 1.297
 gk_Harm1A_Del = 150
 gk_Harm1B_Del = 250
 gk_Harm2A_Del = 370
 gk_Harm2B_Del = 500
endin

instr Cue_57
 puts "Cue_57", 1
 //close all Harm in+out
 gk_Harm1A_VolIn = setTo(gk_Harm1A_VolIn,0,p3)
 gk_Harm1B_VolIn = setTo(gk_Harm1B_VolIn,0,p3)
 gk_Harm2A_VolIn = setTo(gk_Harm2A_VolIn,0,p3)
 gk_Harm2B_VolIn = setTo(gk_Harm2B_VolIn,0,p3)
 gk_Harm1A_VolOut = setTo(gk_Harm1A_VolOut,0,p3)
 gk_Harm1B_VolOut = setTo(gk_Harm1B_VolOut,0,p3)
 gk_Harm2A_VolOut = setTo(gk_Harm2A_VolOut,0,p3)
 gk_Harm2B_VolOut = setTo(gk_Harm2B_VolOut,0,p3)
 //open all Del and Feedback after 1.5 sec
 schedule("Cue_57a",1.5,1)
 //set Harm Prog 19 after 1.5 sec
 schedule("Cue_57b",1.5,1)
 //soft fade in all Harm in+out after 7 sec
 schedule("Cue_57c",7,10)
 //fade in Hala A and fade out Hala C over 3 sec
 schedule("Cue_57d",0,3)
endin

instr Cue_57a
 puts "  Cue_57a", 1
 //open all Del in and Feedback
 gk_Del1A_FbLev = gi_Del_FeedbLev
 gk_Del1B_FbLev = gi_Del_FeedbLev
 gk_Del2B_FbLev = gi_Del_FeedbLev
 gk_Del2A_FbLev = gi_Del_FeedbLev
 gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
 gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
 gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
endin

instr Cue_57b
 puts "  Cue_57b", 1
 //set Harm Prog 19
 gk_Harm1A_Pch = 1.043
 gk_Harm1B_Pch = 0.930
 gk_Harm2A_Pch = 0.939
 gk_Harm2B_Pch = 1.099
 gk_Harm1A_Del = 650
 gk_Harm1B_Del = 1400
 gk_Harm2A_Del = 2300
 gk_Harm2B_Del = 3000
 turnoff
endin

instr Cue_57c
 puts "  Cue_57c", 1
 //soft fade in harm in and out
 iFadeTim = p3
 gk_Harm1A_VolIn linseg 0, iFadeTim/2, 1
 gk_Harm1B_VolIn linseg 0, iFadeTim/2, 1
 gk_Harm2A_VolIn linseg 0, iFadeTim/2, 1
 gk_Harm2B_VolIn linseg 0, iFadeTim/2, 1
 gk_Harm1A_VolOut linseg 0, iFadeTim, 1
 gk_Harm1B_VolOut linseg 0, iFadeTim, 1
```

```
 gk_Harm2A_VolOut linseg 0, iFadeTim, 1
 gk_Harm2B_VolOut linseg 0, iFadeTim, 1
endin

instr Cue_57d
 //fade in Hala A and fade out Hala C over 3 sec
 iFadeTim = p3
 gk_HalaA_VolOut linseg 0, iFadeTim, 1
 gk_HalaC_VolOut linseg 1, iFadeTim, 0
endin

instr Cue_58
 puts "Cue_58", 1
 //all Del in fade out (over 20 sec)
 p3 = 20
 gk_Del1A_VolIn transeg 1, p3, -3, 0
 gk_Del1B_VolIn transeg 1, p3, -3, 0
 gk_Del2A_VolIn transeg 1, p3, -3, 0
 gk_Del2B_VolIn transeg 1, p3, -3, 0
 //live input also fade out
 ga_Mic *= linseg:k(1,p3,0)
 schedule("Cue_58a",0,1)
 //turn on filter
 schedule("Cue_58b",0,1)
 //set filter to initial values (all bands open)
 schedule("Cue_58c",.1,1)
endin

instr Cue_58a
 chnset "Cue 58:\nLive Input muted", "message"
 turnoff
endin

instr Cue_58b
 TurnOnFilter(gi_Filt_NumBands)
 turnoff
endin

instr Cue_58c
 SetFilterInput(gi_Filt_NumBands)
 turnoff
endin

instr Cue_59
 puts "Cue_59", 1
 //set Mtx_13
 schedule("Mtx_13",0,-1)
 //close all Harm in and out
 gk_Harm1A_VolIn = setTo(gk_Harm1A_VolIn,0,p3)
 gk_Harm1B_VolIn = setTo(gk_Harm1B_VolIn,0,p3)
 gk_Harm2A_VolIn = setTo(gk_Harm2A_VolIn,0,p3)
 gk_Harm2B_VolIn = setTo(gk_Harm2B_VolIn,0,p3)
 gk_Harm1A_VolOut = setTo(gk_Harm1A_VolOut,0,p3)
 gk_Harm1B_VolOut = setTo(gk_Harm1B_VolOut,0,p3)
 gk_Harm2A_VolOut = setTo(gk_Harm2A_VolOut,0,p3)
 gk_Harm2B_VolOut = setTo(gk_Harm2B_VolOut,0,p3)
 //then set Harm Prog 20
 schedule("Cue_59a",p3,1)
 //unmute Filter output
 gk_Filt_VolOut = setTo(gk_Filt_VolOut,1,p3)
 //start Filter sequence 2
 schedule("Filt_Seq_2",0,-1)
 //open Hala C output
 gk_HalaC_VolOut = setTo(gk_HalaC_VolOut,1,p3)
 //open live input
 ga_Mic *= linseg:k(0,p3,.5)
 schedule("Cue_59b",0,1)

endin

instr Cue_59a
 gk_Harm1A_Pch = 0.951
 gk_Harm1B_Pch = 0.906
 gk_Harm2A_Pch = 1.051
```

```
 gk_Harm2B_Pch = 1.167
 gk_Harm1A_Del = 150
 gk_Harm1B_Del = 250
 gk_Harm2A_Del = 370
 gk_Harm2B_Del = 500
 turnoff
endin

instr Cue_59b
 chnset "Cue 59:\nLive Input unmuted", "message"
 turnoff
endin

instr Cue_60
 puts "Cue_60", 1
 //open Harm_2B_in
 gk_Harm2B_VolIn = setTo(gk_Harm2B_VolIn,1,p3)
 //open Harm 2 (AB) out
 gk_Harm2A_VolOut = setTo(gk_Harm2A_VolOut,1,p3)
 gk_Harm2B_VolOut = setTo(gk_Harm2B_VolOut,1,p3)
 //open Harm 1 out after 1.5 sec
 schedule("Cue_60a",1.5,1)
endin

instr Cue_60a
 puts "  Cue_60a", 1
 //open Harm 1 out
 gk_Harm1A_VolOut = setTo(gk_Harm1A_VolOut,1,p3)
 gk_Harm1B_VolOut = setTo(gk_Harm1B_VolOut,1,p3)
endin

instr Cue_61
 puts "Cue_61", 1
 //close Harm_2B_in
 gk_Harm2B_VolIn = setTo(gk_Harm2B_VolIn,0,p3)
 //open Harm_2A_in
 gk_Harm2A_VolIn = setTo(gk_Harm2A_VolIn,1,p3)
endin

instr Cue_62
 puts "Cue_62", 1
 //close Harm_2A_in
 gk_Harm2A_VolIn = setTo(gk_Harm2A_VolIn,0,p3)
 //open Harm_1A_in
 gk_Harm1A_VolIn = setTo(gk_Harm1A_VolIn,1,p3)
endin

instr Cue_63
 puts "Cue_63", 1
 //open Harm_1B_in
 gk_Harm1B_VolIn = setTo(gk_Harm1B_VolIn,1,p3)
endin

instr Cue_64
 puts "Cue_64", 1
 //open Harm_2A_in
 gk_Harm2A_VolIn = setTo(gk_Harm2A_VolIn,1,p3)
 //set Harm Prog 21
 gk_Harm1A_Pch = 1.198
 gk_Harm1B_Pch = 0.928
 gk_Harm2A_Pch = 0.865
 gk_Harm2B_Pch = 1.108
 gk_Harm1A_Del = 120
 gk_Harm1B_Del = 210
 gk_Harm2A_Del = 320
 gk_Harm2B_Del = 420
endin

instr Cue_65
 puts "Cue_65", 1
 //open Harm_2B_in
 gk_Harm2B_VolIn = setTo(gk_Harm2B_VolIn,1,p3)
endin

```

```
instr Cue_66
 puts "Cue_66", 1
 //close Harm_1B_in
 gk_Harm1B_VolIn = setTo(gk_Harm1B_VolIn,0,p3)
 //close Harm_2A_in
 gk_Harm2A_VolIn = setTo(gk_Harm2A_VolIn,0,p3)
 //close Harm_2B_in
 gk_Harm2B_VolIn = setTo(gk_Harm2B_VolIn,0,p3)
endin

instr Cue_67
 puts "Cue_67", 1
 //open Harm_1B_in
 gk_Harm1B_VolIn = setTo(gk_Harm1B_VolIn,1,p3)
 //set Harm Prog 22
 gk_Harm1A_Pch = 0.885
 gk_Harm1B_Pch = 1.137
 gk_Harm2A_Pch = 1.297
 gk_Harm2B_Pch = 0.828
 gk_Harm1A_Del = 95
 gk_Harm1B_Del = 170
 gk_Harm2A_Del = 250
 gk_Harm2B_Del = 330
endin

instr Cue_68
 puts "Cue_68", 1
 //open Harm_2A_in
 gk_Harm2A_VolIn = setTo(gk_Harm2A_VolIn,1,p3)
endin

instr Cue_69
 puts "Cue_69", 1
 //open Harm_2B_in
 gk_Harm2B_VolIn = setTo(gk_Harm2B_VolIn,1,p3)
 //set Harm Prog 23
 gk_Harm1A_Pch = 0.793
 gk_Harm1B_Pch = 1.263
 gk_Harm2A_Pch = 0.846
 gk_Harm2B_Pch = 1.198
 gk_Harm1A_Del = 70
 gk_Harm1B_Del = 130
 gk_Harm2A_Del = 180
 gk_Harm2B_Del = 240
endin

instr Cue_70
 puts "Cue_70", 1
 //set Harm Prog 24
 gk_Harm1A_Pch = 0.906
 gk_Harm1B_Pch = 0.766
 gk_Harm2A_Pch = 1.167
 gk_Harm2B_Pch = 1.332
 gk_Harm1A_Del = 40
 gk_Harm1B_Del = 80
 gk_Harm2A_Del = 120
 gk_Harm2B_Del = 160
 turnoff
endin

instr Cue_71
 puts "Cue_71", 1
 //set Harm Prog 25
 gk_Harm1A_Pch = 1.015
 gk_Harm1B_Pch = 0.985
 gk_Harm2A_Pch = 1.031
 gk_Harm2B_Pch = 0.971
 gk_Harm1A_Del = 1
 gk_Harm1B_Del = 1
 gk_Harm2A_Del = 1
 gk_Harm2B_Del = 1
 turnoff
endin

```

```
1554 instr Cue_72
1555  puts "Cue_72", 1
1556  //set Harm Prog 26
1557  gk_Harm1A_Pch = 1.230
1558  gk_Harm1B_Pch = 0.793
1559  gk_Harm2A_Pch = 1.108
1560  gk_Harm2B_Pch = 0.865
1561  turnoff
1562 endin
1563
1564 instr Cue_73
1565  puts "Cue_73", 1
1566  //set Harm Prog 27
1567  gk_Harm1A_Pch = 1.031
1568  gk_Harm1B_Pch = 0.971
1569  gk_Harm2A_Pch = 1.066
1570  gk_Harm2B_Pch = 0.946
1571  turnoff
1572 endin
1573
1574 instr Cue_74
1575  puts "Cue_74", 1
1576  //set Harm Prog 28
1577  gk_Harm1A_Pch = 1.198
1578  gk_Harm1B_Pch = 0.810
1579  gk_Harm2A_Pch = 1.137
1580  gk_Harm2B_Pch = 0.906
1581  turnoff
1582 endin
1583
1584 instr Cue_75
1585  puts "Cue_75", 1
1586  //set Harm Prog 29
1587  gk_Harm1A_Pch = 1.048
1588  gk_Harm1B_Pch = 0.958
1589  gk_Harm2A_Pch = 1.105
1590  gk_Harm2B_Pch = 0.935
1591  turnoff
1592 endin
1593
1594 instr Cue_76
1595  puts "Cue_76", 1
1596  //set Harm Prog 30
1597  gk_Harm1A_Pch = 1.079
1598  gk_Harm1B_Pch = 0.885
1599  gk_Harm2A_Pch = 1.297
1600  gk_Harm2B_Pch = 0.776
1601  turnoff
1602 endin
1603
1604 instr Cue_77
1605  puts "Cue_77", 1
1606  //set Harm Prog 31
1607  gk_Harm1A_Pch = 0.793
1608  gk_Harm1B_Pch = 1.263
1609  gk_Harm2A_Pch = 0.846
1610  gk_Harm2B_Pch = 1.198
1611  gk_Harm1A_Del = 70
1612  gk_Harm1B_Del = 130
1613  gk_Harm2A_Del = 180
1614  gk_Harm2B_Del = 240
1615  turnoff
1616 endin
1617
1618 instr Cue_78
1619  puts "Cue_78", 1
1620  //set Harm Prog 32
1621  gk_Harm1A_Pch = 1.108
1622  gk_Harm1B_Pch = 0.793
1623  gk_Harm2A_Pch = 1.198
1624  gk_Harm2B_Pch = 0.906
1625  gk_Harm1A_Del = 1
1626  gk_Harm1B_Del = 1
1627  gk_Harm2A_Del = 1
```

```
 gk_Harm2B_Del = 1
 turnoff
endin

instr Cue_79
 puts "Cue_79", 1
 //set Harm Prog 33
 gk_Harm1A_Pch = 0.885
 gk_Harm1B_Pch = 1.137
 gk_Harm2A_Pch = 1.297
 gk_Harm2B_Pch = 0.828
 gk_Harm1A_Del = 95
 gk_Harm1B_Del = 170
 gk_Harm2A_Del = 250
 gk_Harm2B_Del = 330
 turnoff
endin

instr Cue_80
 puts "Cue_80", 1
 //set Harm Prog 34
 gk_Harm1A_Pch = 1.066
 gk_Harm1B_Pch = 0.946
 gk_Harm2A_Pch = 1.105
 gk_Harm2B_Pch = 0.925
 gk_Harm1A_Del = 1
 gk_Harm1B_Del = 1
 gk_Harm2A_Del = 1
 gk_Harm2B_Del = 1
 turnoff
endin

instr Cue_81
 puts "Cue_81", 1
 //set Harm Prog 35
 gk_Harm1A_Pch = 1.198
 gk_Harm1B_Pch = 0.928
 gk_Harm2A_Pch = 0.865
 gk_Harm2B_Pch = 1.108
 gk_Harm1A_Del = 120
 gk_Harm1B_Del = 210
 gk_Harm2A_Del = 320
 gk_Harm2B_Del = 420
 turnoff
endin

instr Cue_82
 puts "Cue_82", 1
 //set Harm Prog 36
 gk_Harm1A_Pch = 1.085
 gk_Harm1B_Pch = 0.935
 gk_Harm2A_Pch = 1.105
 gk_Harm2B_Pch = 0.958
 gk_Harm1A_Del = 200
 gk_Harm1B_Del = 350
 gk_Harm2A_Del = 580
 gk_Harm2B_Del = 800
 turnoff
endin

instr Cue_83
 puts "Cue_83", 1
 //set Harm Prog 37
 gk_Harm1A_Pch = 1.031
 gk_Harm1B_Pch = 0.971
 gk_Harm2A_Pch = 1.048
 gk_Harm2B_Pch = 0.958
 gk_Harm1A_Del = 1
 gk_Harm1B_Del = 1
 gk_Harm2A_Del = 1
 gk_Harm2B_Del = 1
 turnoff
endin

```

```
instr Cue_84
 puts "Cue_84", 1
 //set Harm Prog 38
 gk_Harm1A_Pch = 1.031
 gk_Harm1B_Pch = 0.971
 gk_Harm2A_Pch = 1.066
 gk_Harm2B_Pch = 0.946
 gk_Harm1A_Del = 150
 gk_Harm1B_Del = 250
 gk_Harm2A_Del = 370
 gk_Harm2B_Del = 500
 turnoff
endin

instr Cue_85
 puts "Cue_85", 1
 //set Harm Prog 39
 gk_Harm1A_Pch = 1.015
 gk_Harm1B_Pch = 0.985
 gk_Harm2A_Pch = 1.031
 gk_Harm2B_Pch = 0.971
 gk_Harm1A_Del = 150
 gk_Harm1B_Del = 250
 gk_Harm2A_Del = 370
 gk_Harm2B_Del = 500
 schedule("Cue_85a",0,4)
 turnoff
endin

instr Cue_85a
 puts "  Cue_85a", 1
 //fade out Del 2 out and Feedback (over 4 sec)
 iFadeTim = p3
 gk_Del2A_VolOut transeg 1, iFadeTim, -3, 0
 gk_Del2B_VolOut transeg 1, iFadeTim, -3, 0
 gk_Del2A_FbLev linseg gi_Del_FeedbLev, iFadeTim, 0
 gk_Del2B_FbLev linseg gi_Del_FeedbLev, iFadeTim, 0
endin

instr Cue_86
 puts "Cue_86", 1
 //set Mtx_14
 schedule("Mtx_14",p3,-1)
 //set Del 2 Prog 4
 gk_Del2A_DelTim = 9.5
 gk_Del2B_DelTim = 12.7
 //mute Filter output
 gk_Filt_VolOut = setTo(gk_Filt_VolOut,0,p3)
 //mute Hala A and B
 gk_HalaA_VolOut = setTo(gk_HalaA_VolOut,0,p3)
 gk_HalaB_VolOut = setTo(gk_HalaB_VolOut,0,p3)
 //turn off all filter after 3 sec
 schedule("Cue_86c",3,1)
 //set Harm Prog 40
 gk_Harm1A_Pch = 1.137
 gk_Harm1B_Pch = 0.885
 gk_Harm2A_Pch = 1.230
 gk_Harm2B_Pch = 0.793
 gk_Harm1A_Del = 70
 gk_Harm1B_Del = 130
 gk_Harm2A_Del = 180
 gk_Harm2B_Del = 240
 //closes Del 1 out and Feedback
 gk_Del1A_VolOut = setTo(gk_Del1A_VolOut,0,p3)
 gk_Del1B_VolOut = setTo(gk_Del1B_VolOut,0,p3)
 gk_Del1A_FbLev = setTo(gk_Del1A_FbLev,0,p3)
 gk_Del1B_FbLev = setTo(gk_Del1B_FbLev,0,p3)
 //open Del_2B_in
 gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
 //close Del_2B_in after 6 sec
 schedule("Cue_86a",6,1)
 //open Del 2 out after 7 sec
 schedule("Cue_86b",7,1)
endin
```

```
1776
1777 instr Cue_86a
1778  puts "  Cue_86a", 1
1779  gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
1780 endin
1781
1782 instr Cue_86b
1783  puts "  Cue_86b", 1
1784  gk_Del2A_VolOut = setTo(gk_Del2A_VolOut,1,p3)
1785  gk_Del2B_VolOut = setTo(gk_Del2B_VolOut,1,p3)
1786 endin
1787
1788 instr Cue_86c
1789  puts "  Cue_86c", 1
1790  TurnOffFilter
1791  turnoff
1792 endin
1793
1794 instr Cue_87
1795  puts "Cue_87", 1
1796  //open Del 2A in
1797  gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
1798 endin
1799
1800 instr Cue_88
1801  puts "Cue_88", 1
1802  //close Del 2A in
1803  gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,0,p3)
1804  //open Del 2B in
1805  gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
1806 endin
1807
1808 instr Cue_89
1809  puts "Cue_89", 1
1810  //close Del 2B in
1811  gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,0,p3)
1812 endin
1813
1814 instr Cue_90
1815  puts "Cue_90", 1
1816  //close all harm out
1817  gk_Harm1A_VolOut = setTo(gk_Harm1A_VolOut,0,p3)
1818  gk_Harm1B_VolOut = setTo(gk_Harm1B_VolOut,0,p3)
1819  gk_Harm2A_VolOut = setTo(gk_Harm2A_VolOut,0,p3)
1820  gk_Harm2B_VolOut = setTo(gk_Harm2B_VolOut,0,p3)
1821 endin
1822
1823 instr Cue_91
1824  puts "Cue_91", 1
1825  //open all harm out
1826  gk_Harm1A_VolOut = setTo(gk_Harm1A_VolOut,1,p3)
1827  gk_Harm1B_VolOut = setTo(gk_Harm1B_VolOut,1,p3)
1828  gk_Harm2A_VolOut = setTo(gk_Harm2A_VolOut,1,p3)
1829  gk_Harm2B_VolOut = setTo(gk_Harm2B_VolOut,1,p3)
1830 endin
1831
1832 instr Cue_92
1833  puts "Cue_92", 1
1834  //set Del 1 Prog 6
1835  gk_Del1A_DelTim = 7
1836  gk_Del1B_DelTim = 15
1837  //set Harm Prog 41
1838  gk_Harm1A_Pch = 1.079
1839  gk_Harm1B_Pch = 0.928
1840  gk_Harm2A_Pch = 1.167
1841  gk_Harm2B_Pch = 0.846
1842  gk_Harm1A_Del = 1
1843  gk_Harm1B_Del = 1
1844  gk_Harm2A_Del = 1
1845  gk_Harm2B_Del = 1
1846  //open Del 1 out after 2 sec
1847  schedule("Cue_92a",2,1)
1848  turnoff
1849 endin
```

```
1850
1851 instr Cue_92a
1852  puts "  Cue_92a", 1
1853  gk_Del1A_VolOut = setTo(gk_Del1A_VolOut,1,p3)
1854  gk_Del1B_VolOut = setTo(gk_Del1B_VolOut,1,p3)
1855 endin
1856
1857 instr Cue_93
1858  puts "Cue_93", 1
1859  //set Mtx_15
1860  schedule("Mtx_15",0,-1)
1861  //fade out live amplification
1862  schedule("Cue_93a",0,5)
1863  //unmute Hala A and B
1864  gk_HalaA_VolOut = setTo(gk_HalaA_VolOut,1,p3)
1865  gk_HalaB_VolOut = setTo(gk_HalaB_VolOut,1,p3)
1866  //set all Halas to slow speed
1867  iSlow = 1/7
1868  gk_HalaA_Speed = iSlow
1869  gk_HalaB_Speed = iSlow
1870  gk_HalaC_Speed = iSlow
1871  //opens all Del in
1872  gk_Del1A_VolIn = setTo(gk_Del1A_VolIn,1,p3)
1873  gk_Del1B_VolIn = setTo(gk_Del1B_VolIn,1,p3)
1874  gk_Del2A_VolIn = setTo(gk_Del2A_VolIn,1,p3)
1875  gk_Del2B_VolIn = setTo(gk_Del2B_VolIn,1,p3)
1876  //set Harm Prog 42
1877  gk_Harm1A_Pch = 1.031
1878  gk_Harm1B_Pch = 0.958
1879  gk_Harm2A_Pch = 1.066
1880  gk_Harm2B_Pch = 0.946
1881  gk_Harm1A_Del = 200
1882  gk_Harm1B_Del = 350
1883  gk_Harm2A_Del = 580
1884  gk_Harm2B_Del = 800
1885  //start reverb (no input yet)
1886  TurnOnRev
1887 endin
1888
1889 instr Cue_93a
1890  puts "  Cue_93a", 1
1891  gk_LiveAmp_reductDb linseg 0, p3, -50
1892 endin
1893
1894 instr Cue_94
1895  puts "Cue_94", 1
1896  //open reverb in and out
1897  gk_RevA_VolIn = setTo(gk_RevA_VolIn,1,p3)
1898  gk_RevB_VolIn = setTo(gk_RevB_VolIn,1,p3)
1899  gk_RevA_VolOut = setTo(gk_RevA_VolOut,1,p3)
1900  gk_RevB_VolOut = setTo(gk_RevB_VolOut,1,p3)
1901 endin
1902
1903 instr Cue_95
1904  puts "Cue_95", 1
1905  //close reverb in
1906  gk_RevA_VolIn = setTo(gk_RevA_VolIn,0,p3)
1907  gk_RevB_VolIn = setTo(gk_RevB_VolIn,0,p3)
1908 endin
1909
1910 instr Cue_96
1911  puts "Cue_96", 1
1912  //open reverb in
1913  gk_RevA_VolIn = setTo(gk_RevA_VolIn,1,p3)
1914  gk_RevB_VolIn = setTo(gk_RevB_VolIn,1,p3)
1915 endin
1916
1917 instr Cue_97
1918  puts "Cue_97", 1
1919  //all Harm out fade out then fade in (each 3 sec)
1920  iFadeTim = 3
1921  p3 = 2*iFadeTim
1922  gk_Harm1A_VolOut transeg 1, iFadeTim, -3, 0, iFadeTim, 3, 1
1923  gk_Harm1B_VolOut transeg 1, iFadeTim, -3, 0, iFadeTim, 3, 1
```

```
 gk_Harm2A_VolOut transeg 1, iFadeTim, -3, 0, iFadeTim, 3, 1
 gk_Harm2B_VolOut transeg 1, iFadeTim, -3, 0, iFadeTim, 3, 1
endin

instr Cue_98
 puts "Cue_98", 1
 //all Harm out fade out over 7 sec
 iFadeTim = 7
 p3 = iFadeTim
 gk_Harm1A_VolOut transeg 1, iFadeTim, -3, 0
 gk_Harm1B_VolOut transeg 1, iFadeTim, -3, 0
 gk_Harm2A_VolOut transeg 1, iFadeTim, -3, 0
 gk_Harm2B_VolOut transeg 1, iFadeTim, -3, 0
endin


/*                                                                              */
/********************************************************************************/
/********************************************************************************/



/********************************************************************************/
/********************************************************************************/
/*                                                                              */
/*                              MIDI CONTROLLER                                 */

instr Midi

iPortTim = .1 ;sec

gk_DelOut_midiDb chnget "Del_out_midi"
gk_DelOut_midiDb port gk_DelOut_midiDb, iPortTim
gk_HarmOut_midiDb chnget "Harm_out_midi"
gk_HarmOut_midiDb port gk_HarmOut_midiDb, iPortTim
gk_FiltOut_midiDb chnget "Filt_out_midi"
gk_FiltOut_midiDb port gk_FiltOut_midiDb, iPortTim
gk_RevOut_midiDb chnget "Rev_out_midi"
gk_RevOut_midiDb port gk_RevOut_midiDb, iPortTim
gk_HalaAOut_midiDb chnget "Hala_A_out_midi"
gk_HalaAOut_midiDb port gk_HalaAOut_midiDb, iPortTim
gk_HalaBOut_midiDb chnget "Hala_B_out_midi"
gk_HalaBOut_midiDb port gk_HalaBOut_midiDb, iPortTim
gk_HalaCOut_midiDb chnget "Hala_C_out_midi"
gk_HalaCOut_midiDb port gk_HalaCOut_midiDb, iPortTim
gk_SumOut_midiDb chnget "Sum_out_midi"
gk_SumOut_midiDb port gk_SumOut_midiDb, iPortTim
gk_MicA_midiDb chnget "micA_gain_midi"
gk_MicA_midiDb port gk_MicA_midiDb, iPortTim
gk_MicB_midiDb chnget "micB_gain_midi"
gk_MicB_midiDb port gk_MicB_midiDb, iPortTim
gk_LiveAmp_midiDb chnget "live_amp_midi"
gk_LiveAmp_midiDb += gk_LiveAmp_reductDb
gk_LiveAmp_midiDb port gk_LiveAmp_midiDb, iPortTim
 ;hala speed can be modified by knobs (0=normal, positive=faster, negative=slower)
gk_HalaASpeed_midi chnget "Hala_A_speed_midi"
gk_HalaBSpeed_midi chnget "Hala_B_speed_midi"
gk_HalaCSpeed_midi chnget "Hala_C_speed_midi"
 ;reverb feedback level can be subtracted (max -0.1) from 0.99
gk_RevReduct_midi chnget "Rev_reduct_midi"
 ;filter Q (make sure not to be < 1)
gk_Filt_Q chnget "filt_q"
gk_Filt_Q limit gk_Filt_Q, 1, 24

endin


/*                                                                              */
/********************************************************************************/
/********************************************************************************/
```

```
1998
1999 /*****************************************************************************/
2000 /*****************************************************************************/
2001 /*                                                                           */
2002 /*                               MATRIX SETTINGS                             */
2003
2004
2005 instr Mtx_1
2006
2007  puts "Mtx_1", 1
2008  chnset 1, "show_mtx"
2009
2010  ga_Harm_in = ga_Del_out
2011  ga_Chn1_in = ga_Harm1A_out
2012  ga_Chn2_in = ga_Harm1B_out
2013  ga_Chn3_in = ga_Harm2A_out
2014  ga_Chn4_in = ga_Harm2B_out
2015  ga_Chn5_in = 0
2016  ga_Chn6_in = 0
2017  ga_Filt_in = 0
2018  ga_Rev_in = 0
2019  ga_HalaA_in = 0
2020  ga_HalaB_in = 0
2021  ga_HalaC_in = 0
2022  TurOffOtherMtxs gS_Mtxs, "Mtx_1"
2023
2024 endin
2025
2026 instr Mtx_2
2027
2028  puts "Mtx_2", 1
2029  chnset 2, "show_mtx"
2030
2031  ga_Harm_in = ga_Del_out
2032  ga_Chn1_in = ga_Harm1A_out
2033  ga_Chn2_in = 0
2034  ga_Chn3_in = 0
2035  ga_Chn4_in = ga_Harm2B_out
2036  ga_Chn5_in = ga_Harm2A_out
2037  ga_Chn6_in = ga_Harm1B_out
2038  ga_Filt_in = 0
2039  ga_Rev_in = 0
2040  ga_HalaA_in = 0
2041  ga_HalaB_in = 0
2042  ga_HalaC_in = 0
2043  TurOffOtherMtxs gS_Mtxs, "Mtx_2"
2044
2045 endin
2046
2047 instr Mtx_3
2048
2049  puts "Mtx_3", 1
2050  chnset 3, "show_mtx"
2051
2052  ga_Harm_in = ga_Del_out
2053  ga_Chn1_in = 0
2054  ga_Chn2_in = ga_Harm1B_out
2055  ga_Chn3_in = ga_Harm2B_out
2056  ga_Chn4_in = ga_Harm2A_out
2057  ga_Chn5_in = ga_Harm1A_out
2058  ga_Chn6_in = 0
2059  ga_Filt_in = 0
2060  ga_Rev_in = 0
2061  ga_HalaA_in = 0
2062  ga_HalaB_in = 0
2063  ga_HalaC_in = 0
2064  TurOffOtherMtxs gS_Mtxs, "Mtx_3"
2065
2066 endin
2067
2068 instr Mtx_4
2069
2070  puts "Mtx_4", 1
2071  chnset 4, "show_mtx"
```

```
2072
2073  ga_Harm_in = ga_Del_out
2074  ga_Chn1_in = ga_Harm2B_out
2075  ga_Chn2_in = ga_Harm1A_out
2076  ga_Chn3_in = 0
2077  ga_Chn4_in = ga_Harm1B_out
2078  ga_Chn5_in = ga_Harm2A_out
2079  ga_Chn6_in = 0
2080  ga_Filt_in = 0
2081  ga_Rev_in = 0
2082  ga_HalaA_in = 0
2083  ga_HalaB_in = 0
2084  ga_HalaC_in = 0
2085  TurOffOtherMtxs gS_Mtxs, "Mtx_4"
2086
2087 endin
2088
2089 instr Mtx_5
2090
2091  puts "Mtx_5", 1
2092  chnset 5, "show_mtx"
2093
2094  ga_Harm_in = ga_Del_out
2095  ga_Chn1_in = 0
2096  ga_Chn2_in = ga_Harm2B_out
2097  ga_Chn3_in = ga_Harm2A_out
2098  ga_Chn4_in = ga_Harm1B_out
2099  ga_Chn5_in = ga_Harm1A_out
2100  ga_Chn6_in = 0
2101  ga_Filt_in = 0
2102  ga_Rev_in = 0
2103  ga_HalaA_in = 0
2104  ga_HalaB_in = 0
2105  ga_HalaC_in = 0
2106  TurOffOtherMtxs gS_Mtxs, "Mtx_5"
2107
2108 endin
2109
2110 instr Mtx_6
2111
2112  puts "Mtx_6", 1
2113  chnset 6, "show_mtx"
2114
2115  ga_Harm_in = ga_Del_out
2116  ga_Chn1_in = ga_Harm2A_out
2117  ga_Chn2_in = 0
2118  ga_Chn3_in = ga_Harm2B_out
2119  ga_Chn4_in = ga_Harm1B_out
2120  ga_Chn5_in = ga_Harm1A_out
2121  ga_Chn6_in = 0
2122  ga_Filt_in = 0
2123  ga_Rev_in = 0
2124  ga_HalaA_in = 0
2125  ga_HalaB_in = 0
2126  ga_HalaC_in = 0
2127  TurOffOtherMtxs gS_Mtxs, "Mtx_6"
2128
2129 endin
2130
2131 instr Mtx_7
2132
2133  puts "Mtx_7", 1
2134  chnset 7, "show_mtx"
2135
2136  ga_Harm_in = ga_Del_out
2137  ga_Chn1_in = ga_Harm1A_out
2138  ga_Chn2_in = ga_Harm1B_out
2139  ga_Chn3_in = 0
2140  ga_Chn4_in = ga_Harm2A_out
2141  ga_Chn5_in = ga_Harm2B_out
2142  ga_Chn6_in = 0
2143  ga_Filt_in = 0
2144  ga_Rev_in = 0
2145  ga_HalaA_in = 0
```

```
 ga_HalaB_in = 0
 ga_HalaC_in = 0
 TurOffOtherMtxs gS_Mtxs, "Mtx_7"

endin

instr Mtx_8

 puts "Mtx_8", 1
 chnset 8, "show_mtx"

 ga_Harm_in = ga_Mic
 ga_Chn1_in = ga_Harm1A_out
 ga_Chn2_in = ga_Harm1B_out
 ga_Chn3_in = ga_Harm2A_out
 ga_Chn4_in = ga_Harm2B_out
 ga_Chn5_in = 0
 ga_Chn6_in = 0
 ga_Filt_in = 0
 ga_Rev_in = 0
 ga_HalaA_in = 0
 ga_HalaB_in = 0
 ga_HalaC_in = 0
 TurOffOtherMtxs gS_Mtxs, "Mtx_8"

endin

instr Mtx_9

 puts "Mtx_9", 1
 chnset 9, "show_mtx"

 ga_Harm_in = ga_Del1_out
 ga_Chn1_in = ga_Harm1A_out
 ga_Chn2_in = ga_Harm2A_out
 ga_Chn3_in = 0
 ga_Chn4_in = ga_Harm1B_out
 ga_Chn5_in = ga_Harm2B_out
 ga_Chn6_in = 0
 ga_Filt_in = ga_Del2_out
 ga_Rev_in = 0
 ga_HalaA_in = ga_Filt_out
 ga_HalaB_in = ga_Filt_out
 ga_HalaC_in = 0
 TurOffOtherMtxs gS_Mtxs, "Mtx_9"

endin

instr Mtx_10

 puts "Mtx_10", 1
 chnset 10, "show_mtx"

 ga_Harm_in = ga_Mic
 ga_Chn1_in = ga_Harm1A_out
 ga_Chn2_in = ga_Harm1B_out
 ga_Chn3_in = ga_Harm1A_out
 ga_Chn4_in = ga_Harm2A_out
 ga_Chn5_in = ga_Harm2B_out
 ga_Chn6_in = ga_Harm1B_out
 ga_Filt_in = ga_Del2_out
 ga_Rev_in = 0
 ga_HalaA_in = ga_Filt_out
 ga_HalaB_in = ga_Filt_out
 ga_HalaC_in = 0
 TurOffOtherMtxs gS_Mtxs, "Mtx_10"

endin

instr Mtx_11

 puts "Mtx_11", 1
 chnset 11, "show_mtx"

```

```
2220  ga_Harm_in = ga_Del_out
2221  ga_Chn1_in = ga_Harm1A_out
2222  ga_Chn2_in = ga_Harm1B_out
2223  ga_Chn3_in = 0
2224  ga_Chn4_in = ga_Harm2A_out
2225  ga_Chn5_in = ga_Harm2B_out
2226  ga_Chn6_in = 0
2227  ga_Filt_in = 0
2228  ga_Rev_in = 0
2229  ga_HalaA_in = 0
2230  ga_HalaB_in = 0
2231  ga_HalaC_in = 0
2232  TurOffOtherMtxs gS_Mtxs, "Mtx_11"
2233
2234 endin
2235
2236 instr Mtx_12
2237
2238  puts "Mtx_12", 1
2239  chnset 12, "show_mtx"
2240
2241  ga_Harm_in = ga_Del_out
2242  ga_Chn1_in = 0
2243  ga_Chn2_in = 0
2244  ga_Chn3_in = ga_Harm2A_out
2245  ga_Chn4_in = ga_Harm1B_out
2246  ga_Chn5_in = ga_Harm1A_out
2247  ga_Chn6_in = ga_Harm2B_out
2248  ga_Filt_in = 0
2249  ga_Rev_in = 0
2250  ga_HalaA_in = ga_Del_out
2251  ga_HalaB_in = ga_Del_out
2252  ga_HalaC_in = ga_Del_out
2253  TurOffOtherMtxs gS_Mtxs, "Mtx_12"
2254
2255 endin
2256
2257 instr Mtx_13
2258
2259  puts "Mtx_13", 1
2260  chnset 13, "show_mtx"
2261
2262  ga_Harm_in = ga_Mic
2263  ga_Chn1_in = 0
2264  ga_Chn2_in = 0
2265  ga_Chn3_in = ga_Harm1A_out
2266  ga_Chn4_in = ga_Harm2A_out
2267  ga_Chn5_in = ga_Harm2B_out
2268  ga_Chn6_in = ga_Harm1B_out
2269  ga_Filt_in = ga_Del_out
2270  ga_Rev_in = 0
2271  ga_HalaA_in = ga_Filt_out
2272  ga_HalaB_in = ga_Filt_out
2273  ga_HalaC_in = ga_Mic
2274  TurOffOtherMtxs gS_Mtxs, "Mtx_13"
2275
2276 endin
2277
2278 instr Mtx_14
2279
2280  puts "Mtx_14", 1
2281  chnset 14, "show_mtx"
2282
2283  ga_Harm_in = ga_Del_out
2284  ga_Chn1_in = 0
2285  ga_Chn2_in = 0
2286  ga_Chn3_in = ga_Harm1A_out
2287  ga_Chn4_in = ga_Harm1B_out
2288  ga_Chn5_in = ga_Harm2A_out
2289  ga_Chn6_in = ga_Harm2B_out
2290  ga_Filt_in = 0
2291  ga_Rev_in = 0
2292  ga_HalaA_in = 0
2293  ga_HalaB_in = 0
```

```
 ga_HalaC_in = ga_Mic
 TurOffOtherMtxs gS_Mtxs, "Mtx_14"

endin

instr Mtx_15

 puts "Mtx_15", 1
 chnset 15, "show_mtx"

 ga_Harm_in = ga_Del_out
 ga_Chn1_in = ga_RevA_out
 ga_Chn2_in = ga_RevA_out
 ga_Chn3_in = 0
 ga_Chn4_in = ga_RevB_out
 ga_Chn5_in = ga_RevB_out
 ga_Chn6_in = 0
 ga_Filt_in = 0
 ga_Rev_in = ga_Del_out
 ga_HalaA_in = ga_Harm_out
 ga_HalaB_in = ga_Harm_out
 ga_HalaC_in = ga_Harm_out
 TurOffOtherMtxs gS_Mtxs, "Mtx_15"

endin


/*                                                                              */
/********************************************************************************/
/********************************************************************************/


/********************************************************************************/
/********************************************************************************/
/*                                                                              */
/*                                DEVICE UNITS                                  */

/*** INPUT ***/

instr Input

 kMicA_input_chn chnget "micA_input_chn"
 kMicB_input_chn chnget "micB_input_chn"
 kSelect chnget "mic_select"

 aMicA_pre inch kMicA_input_chn
 aMicB_pre inch kMicB_input_chn
 ga_MicA = aMicA_pre * ampdb(gk_MicA_midiDb)
 ga_MicB = aMicB_pre * ampdb(gk_MicB_midiDb)

 if kSelect == 0 then
  ga_Mic = ga_MicA

 elseif kSelect == 1 then
  ga_Mic = ga_MicB

 else
  ga_Mic = ga_MicA+ga_MicB

 endif

endin


/*** DELAY ***/

instr Del_1A

 aInput = ga_Mic
 kVolIn = gk_Del1A_VolIn
 kVolOut = gk_Del1A_VolOut
```

```
 kDelTim = gk_Del1A_DelTim ;sec
 kDelTim *= 1000 ;ms
 kFbLev = gk_Del1A_FbLev

 aInput = aInput * kVolIn
 aDelay init 0
 aFeedback = aDelay * kFbLev
 aDelay vdelay aInput+aFeedback, a(kDelTim), gi_Del_MaxDelTim*1000
 aOut = aDelay * kVolOut

 ga_Del1A_out = aOut * ampdb(gk_DelOut_midiDb)

endin

instr Del_1B

 aInput = ga_Mic
 kVolIn = gk_Del1B_VolIn
 kVolOut = gk_Del1B_VolOut
 kDelTim = gk_Del1B_DelTim ;sec
 kDelTim *= 1000 ;ms
 kFbLev = gk_Del1B_FbLev

 aInput = aInput * kVolIn
 aDelay init 0
 aFeedback = aDelay * kFbLev
 aDelay vdelay aInput+aFeedback, a(kDelTim), gi_Del_MaxDelTim*1000
 aOut = aDelay * kVolOut

 ga_Del1B_out = aOut * ampdb(gk_DelOut_midiDb)

endin

instr Del_2A

 aInput = ga_Mic
 kVolIn = gk_Del2A_VolIn
 kVolOut = gk_Del2A_VolOut
 kDelTim = gk_Del2A_DelTim ;sec
 kDelTim *= 1000 ;ms
 kFbLev = gk_Del2A_FbLev

 aInput = aInput * kVolIn
 aDelay init 0
 aFeedback = aDelay * kFbLev
 aDelay vdelay aInput+aFeedback, a(kDelTim), gi_Del_MaxDelTim*1000
 aOut = aDelay * kVolOut

 ga_Del2A_out = aOut * ampdb(gk_DelOut_midiDb)

endin

instr Del_2B

 aInput = ga_Mic
 kVolIn = gk_Del2B_VolIn
 kVolOut = gk_Del2B_VolOut
 kDelTim = gk_Del2B_DelTim ;sec
 kDelTim *= 1000 ;ms
 kFbLev = gk_Del2B_FbLev

 aInput = aInput * kVolIn
 aDelay init 0
 aFeedback = aDelay * kFbLev
 aDelay vdelay aInput+aFeedback, a(kDelTim), gi_Del_MaxDelTim*1000
 aOut = aDelay * kVolOut

 ga_Del2B_out = aOut * ampdb(gk_DelOut_midiDb)

endin

instr Del_sums

 ga_Del1_out = ga_Del1A_out + ga_Del1B_out
```

```
 ga_Del2_out = ga_Del2A_out + ga_Del2B_out
 ga_Del_out = ga_Del1_out + ga_Del2_out

endin



/*** HARMONIZER ***/

instr Harm_1A

 aInput = ga_Harm_in
 kVolIn = gk_Harm1A_VolIn
 kVolOut = gk_Harm1A_VolOut
 kPch = gk_Harm1A_Pch
 kDel = gk_Harm1A_Del ;ms!

 aInput = aInput * kVolIn

 //transposition (pitch shifting)
 fInput pvsanal aInput, giFFTsize, giFFThopsize, giFFTsize, 1
 fTransp pvscale fInput, kPch
 aTransp pvsynth fTransp

 //variable delay
 aDel vdelay3 aTransp, a(kDel), gi_Harm_MaxDelTim

 //output
 aOut = aDel * kVolOut
 ga_Harm1A_out = aOut * ampdb(gk_HarmOut_midiDb)

endin

instr Harm_1B

 aInput = ga_Harm_in
 kVolIn = gk_Harm1B_VolIn
 kVolOut = gk_Harm1B_VolOut
 kPch = gk_Harm1B_Pch
 kDel = gk_Harm1B_Del ;ms!

 aInput = aInput * kVolIn

 //transposition (pitch shifting)
 fInput pvsanal aInput, giFFTsize, giFFThopsize, giFFTsize, 1
 fTransp pvscale fInput, kPch
 aTransp pvsynth fTransp

 //variable delay
 aDel vdelay3 aTransp, a(kDel), gi_Harm_MaxDelTim

 //output
 aOut = aDel * kVolOut
 ga_Harm1B_out = aOut * ampdb(gk_HarmOut_midiDb)

endin

instr Harm_2A

 aInput = ga_Harm_in
 kVolIn = gk_Harm2A_VolIn
 kVolOut = gk_Harm2A_VolOut
 kPch = gk_Harm2A_Pch
 kDel = gk_Harm2A_Del ;ms!

 aInput = aInput * kVolIn

 //transposition (pitch shifting)
 fInput pvsanal aInput, giFFTsize, giFFThopsize, giFFTsize, 1
 fTransp pvscale fInput, kPch
 aTransp pvsynth fTransp

 //variable delay
 aDel vdelay3 aTransp, a(kDel), gi_Harm_MaxDelTim
```

```
2516
2517  //output
2518  aOut = aDel * kVolOut
2519  ga_Harm2A_out = aOut * ampdb(gk_HarmOut_midiDb)
2520
2521 endin
2522
2523 instr Harm_2B
2524
2525  aInput = ga_Harm_in
2526  kVolIn = gk_Harm2B_VolIn
2527  kVolOut = gk_Harm2B_VolOut
2528  kPch = gk_Harm2B_Pch
2529  kDel = gk_Harm2B_Del ;ms!
2530
2531  aInput = aInput * kVolIn
2532
2533  //transposition (pitch shifting)
2534  fInput pvsanal aInput, giFFTsize, giFFThopsize, giFFTsize, 1
2535  fTransp pvscale fInput, kPch
2536  aTransp pvsynth fTransp
2537
2538  //variable delay
2539  aDel vdelay3 aTransp, a(kDel), gi_Harm_MaxDelTim
2540
2541  //output
2542  aOut = aDel * kVolOut
2543  ga_Harm2B_out = aOut * ampdb(gk_HarmOut_midiDb)
2544
2545 endin
2546
2547 instr Harm_sums
2548
2549  ga_Harm1_out = ga_Harm1A_out + ga_Harm1B_out
2550  ga_Harm2_out = ga_Harm2A_out + ga_Harm2B_out
2551  ga_Harm_out = ga_Harm1_out + ga_Harm2_out
2552
2553 endin
2554
2555
2556
2557 /*** FILTER ***/
2558
2559 instr Filt_Seq_1
2560
2561  kndx init 0
2562  kTime init 0
2563  kFiltSeq[] = gk_Filt_Seq_1
2564  iFirstProg = 1
2565
2566  if kTime <= 0 then
2567   event "i", "ReadFiltProg", 0, 0, iFirstProg+kndx
2568   kTime = kFiltSeq[kndx]
2569   kndx += 1
2570   if kndx == lenarray(kFiltSeq) then
2571    printks "  Filt_Seq_1 turned off\n", 0
2572    turnoff
2573   endif
2574  endif
2575
2576  kTime -= 1/kr
2577
2578 endin
2579
2580 instr Filt_Seq_2
2581
2582  kndx init 0
2583  kTime init 0
2584  kFiltSeq[] = gk_Filt_Seq_2
2585  iFirstProg = 15
2586
2587  if kTime <= 0 then
2588   event "i", "ReadFiltProg", 0, 0, iFirstProg+kndx
2589   kTime = kFiltSeq[kndx]
```

```
  kndx += 1
  if kndx == lenarray(kFiltSeq) then
   printks "  Filt_Seq_2 turned off\n", 0
   turnoff
  endif
 endif

 kTime -= 1/kr

endin

instr ReadFiltProg

 iProg = p4 ;1, 2, 3, ...
 ;print iProg

 indx = 0
 while indx < gi_Filt_NumBands do
  iDb_A table indx + gi_Filt_NumBands * (iProg-1), giFilt_A
  iDb_B table indx + gi_Filt_NumBands * (iProg-1), giFilt_B
  chnset iDb_A, sprintf("Filt_A_%d",indx+1)
  chnset iDb_B, sprintf("Filt_B_%d",indx+1)
  indx += 1
 od

endin

instr Filt_A

 iBand = p4
 S_chnl sprintf "Filt_A_%d", iBand

 //midi pitch one tone below the first band
 iBasPch = 34

 iFreq mtof iBasPch + iBand*2
 kDb chnget S_chnl
 kDb port kDb, gi_Filt_FadeTim
 ;aFilt mode ga_Filt_in*ampdb(kDb), iFreq, gk_Filt_Q
 aFilt zdf_2pole ga_Filt_in*ampdb(kDb), iFreq, gk_Filt_Q, 2, 0

 chnmix aFilt, "filt_A_collect"

endin

instr Filt_B

 iBand = p4
 S_chnl sprintf "Filt_B_%d", iBand

 //midi pitch one tone below the first band
 iBasPch = 33

 iFreq mtof iBasPch + iBand*2
 kDb chnget S_chnl
 kDb port kDb, gi_Filt_FadeTim
 ;aFilt mode ga_Filt_in*ampdb(kDb), iFreq, gk_Filt_Q
 aFilt zdf_2pole ga_Filt_in*ampdb(kDb), iFreq, gk_Filt_Q, 2, 0

 chnmix aFilt, "filt_B_collect"

endin

instr Filt

 iFiltAtt = 10 ;use 50 for mode filter
 ga_FiltA_out chnget "filt_A_collect"
 ga_FiltB_out chnget "filt_B_collect"
 ga_Filt_out = ga_FiltA_out/iFiltAtt + ga_FiltB_out/iFiltAtt
 ga_Filt_out *= gk_Filt_VolOut  * ampdb(gk_FiltOut_midiDb)
 chnclear "filt_A_collect"
 chnclear "filt_B_collect"

 //report running instances of Filt_A and _B
```

```
 Report "Filt_A"
 Report "Filt_B"

endin



/*** REVERB ***/

instr Rev_AB

 iFB_lev = .99
 kFB_lev = iFB_lev + gk_RevReduct_midi
 aIn_A = ga_Rev_in * gk_RevA_VolIn
 aIn_B = ga_Rev_in * gk_RevB_VolIn

 aOut_A, aOut_B reverbsc aIn_A, aIn_B, kFB_lev, sr/2

 ga_RevA_out = aOut_A * gk_RevA_VolOut * ampdb(gk_RevOut_midiDb)
 ga_RevB_out = aOut_B * gk_RevB_VolOut * ampdb(gk_RevOut_midiDb)

endin



/*** HALAFON ***/

instr Hala_A

 aIn = ga_HalaA_in
 ga_HalaA_out = aIn * gk_HalaA_VolOut * ampdb(gk_HalaAOut_midiDb)
 kSpeed = gk_HalaA_Speed * 2^gk_HalaASpeed_midi
 chnset kSpeed, "HalaA_speed_disp"
 kDir = gk_HalaA_Dir

 kAzi phasor kSpeed*kDir
 kAzi *= 360
 aOut[] vbap ga_HalaA_out, kAzi, 0, gk_Hala_Spread, gi_Setup
 ga_Out = ga_Out + aOut

endin

instr Hala_B

 aIn = ga_HalaB_in
 ga_HalaB_out = aIn * gk_HalaB_VolOut * ampdb(gk_HalaBOut_midiDb)
 kSpeed = gk_HalaB_Speed * 2^gk_HalaBSpeed_midi
 chnset kSpeed, "HalaB_speed_disp"
 kDir = gk_HalaB_Dir

 kAzi phasor kSpeed*kDir
 kAzi *= 360
 aOut[] vbap ga_HalaB_out, kAzi, 0, gk_Hala_Spread, gi_Setup
 ga_Out = ga_Out + aOut

endin

instr Hala_C

 aIn = ga_HalaC_in
 ga_HalaC_out = aIn * gk_HalaC_VolOut * ampdb(gk_HalaCOut_midiDb)
 kSpeed = gk_HalaC_Speed * 2^gk_HalaCSpeed_midi
 chnset kSpeed, "HalaC_speed_disp"
 kDir = gk_HalaC_Dir

 kAzi phasor kSpeed*kDir
 kAzi *= 360
 aOut[] vbap ga_HalaC_out, kAzi, 0, gk_Hala_Spread, gi_Setup
 ga_Out = ga_Out + aOut

endin



```

```
2738 /*** DIRECT OUT AS IN SCORE ***/
2739
2740 instr Chn_1
2741
2742  aIn = ga_Chn1_in
2743  iAzi = -30
2744
2745  aOut[] vbap aIn, iAzi, 0, 0, gi_Setup
2746  ga_Out = ga_Out + aOut
2747
2748 endin
2749
2750 instr Chn_2
2751
2752  aIn = ga_Chn2_in
2753  iAzi = 30
2754
2755  aOut[] vbap aIn, iAzi, 0, 0, gi_Setup
2756  ga_Out = ga_Out + aOut
2757
2758 endin
2759
2760 instr Chn_3
2761
2762  aIn = ga_Chn3_in
2763  iAzi = 90
2764
2765  aOut[] vbap aIn, iAzi, 0, 0, gi_Setup
2766  ga_Out = ga_Out + aOut
2767
2768 endin
2769
2770 instr Chn_4
2771
2772  aIn = ga_Chn4_in
2773  iAzi = 150
2774
2775  aOut[] vbap aIn, iAzi, 0, 0, gi_Setup
2776  ga_Out = ga_Out + aOut
2777
2778 endin
2779
2780 instr Chn_5
2781
2782  aIn = ga_Chn5_in
2783  iAzi = -150
2784
2785  aOut[] vbap aIn, iAzi, 0, 0, gi_Setup
2786  ga_Out = ga_Out + aOut
2787
2788 endin
2789
2790 instr Chn_6
2791
2792  aIn = ga_Chn6_in
2793  iAzi = -90
2794
2795  aOut[] vbap aIn, iAzi, 0, 0, gi_Setup
2796  ga_Out = ga_Out + aOut
2797
2798 endin
2799
2800
2801 /*                                                                      */
2802 /**********************************************************************/
2803 /**********************************************************************/
2804
2805
2806
2807 /**********************************************************************/
2808 /**********************************************************************/
2809 /*                                                                      */
2810 /*                                OUTPUT                                */
2811
```

```
2812
2813 instr Output
2814
2815  kLiveAmp chnget "live_amp_select"
2816  if (kLiveAmp == 1) && (chnget:k("count") > 0) then
2817   ga_Out[0] = ga_Out[0] + ga_Mic*ampdb(gk_LiveAmp_midiDb)
2818   ga_Out[1] = ga_Out[1] + ga_Mic*ampdb(gk_LiveAmp_midiDb)
2819  endif
2820
2821  ga_Out *= ampdb(gk_SumOut_midiDb)
2822
2823  //simple way for output on the first 4, 6 or 8 channels
2824  out ga_Out
2825
2826  //alternative way -- uncomment which signals are active
2827  ;outch gi_OutChn_1, ga_Out[0]
2828  ;outch gi_OutChn_2, ga_Out[1]
2829  ;outch gi_OutChn_3, ga_Out[2]
2830  ;outch gi_OutChn_4, ga_Out[3]
2831  ;outch gi_OutChn_5, ga_Out[4]
2832  ;outch gi_OutChn_6, ga_Out[5]
2833  ;outch gi_OutChn_7, ga_Out[6]
2834  ;outch gi_OutChn_8, ga_Out[7]
2835
2836  //show audio signals
2837  kTrig metro 10
2838  CsQtMeter("micA_in_disp","micA_in_over_disp",ga_MicA,kTrig)
2839  CsQtMeter("micB_in_disp","micB_in_over_disp",ga_MicB,kTrig)
2840  CsQtMeter("micSum_in_disp","micSum_in_over_disp",ga_Mic,kTrig)
2841  CsQtMeter("Del_1_disp","Del_1_over_disp",ga_Del1_out,kTrig)
2842  CsQtMeter("Del_2_disp","Del_2_over_disp",ga_Del2_out,kTrig)
2843  CsQtMeter("Harm_1_disp","Harm_1_over_disp",ga_Harm1_out,kTrig)
2844  CsQtMeter("Harm_2_disp","Harm_2_over_disp",ga_Harm2_out,kTrig)
2845  CsQtMeter("Rev_disp","Rev_over_disp",(ga_RevA_out+ga_RevB_out)/2,kTrig)
2846  CsQtMeter("Filt_disp","Filt_over_disp",ga_Filt_out,kTrig)
2847  CsQtMeter("Hala_A_disp","Hala_A_over_disp",ga_HalaA_out,kTrig)
2848  CsQtMeter("Hala_B_disp","Hala_B_over_disp",ga_HalaB_out,kTrig)
2849  CsQtMeter("Hala_C_disp","Hala_C_over_disp",ga_HalaC_out,kTrig)
2850  CsQtMeter("out_1_disp","out_1_over_disp",ga_Out[0],kTrig)
2851  CsQtMeter("out_2_disp","out_2_over_disp",ga_Out[1],kTrig)
2852  CsQtMeter("out_3_disp","out_3_over_disp",ga_Out[2],kTrig)
2853  CsQtMeter("out_4_disp","out_4_over_disp",ga_Out[3],kTrig)
2854  if gi_Setup > 1 then
2855   CsQtMeter("out_5_disp","out_5_over_disp",ga_Out[4],kTrig)
2856   CsQtMeter("out_6_disp","out_6_over_disp",ga_Out[5],kTrig)
2857  endif
2858  if gi_Setup > 2 then
2859   CsQtMeter("out_7_disp","out_7_over_disp",ga_Out[6],kTrig)
2860   CsQtMeter("out_8_disp","out_8_over_disp",ga_Out[7],kTrig)
2861  endif
2862
2863  //clear global audio array
2864  ga_Out = ga_Clear
2865
2866 endin
2867
2868
2869 /*                                                                            */
2870 /****************************************************************************/
2871 /****************************************************************************/
2872
2873
2874
2875 /****************************************************************************/
2876 /****************************************************************************/
2877 /*                                                                          */
2878 /*                              UNIT DISPLAY                                */
2879
2880 instr Show
2881
2882  chnset gk_Del1A_VolIn, "Del_1A_in"
2883  chnset gk_Del1B_VolIn, "Del_1B_in"
2884  chnset gk_Del2A_VolIn, "Del_2A_in"
2885  chnset gk_Del2B_VolIn, "Del_2B_in"
```

```
 chnset gk_Del1A_VolOut, "Del_1A_out"
 chnset gk_Del1B_VolOut, "Del_1B_out"
 chnset gk_Del2A_VolOut, "Del_2A_out"
 chnset gk_Del2B_VolOut, "Del_2B_out"
 chnset gk_Harm1A_VolIn, "Harm_1A_in"
 chnset gk_Harm1B_VolIn, "Harm_1B_in"
 chnset gk_Harm2A_VolIn, "Harm_2A_in"
 chnset gk_Harm2B_VolIn, "Harm_2B_in"
 chnset gk_Harm1A_VolOut, "Harm_1A_out"
 chnset gk_Harm1B_VolOut, "Harm_1B_out"
 chnset gk_Harm2A_VolOut, "Harm_2A_out"
 chnset gk_Harm2B_VolOut, "Harm_2B_out"
 chnset gk_Filt_VolOut, "Filt_out"
 chnset gk_RevA_VolIn, "Rev_A_in"
 chnset gk_RevB_VolIn, "Rev_B_in"
 chnset gk_RevA_VolOut, "Rev_A_out"
 chnset gk_RevB_VolOut, "Rev_B_out"
 chnset gk_HalaA_VolOut, "Hala_A"
 chnset gk_HalaB_VolOut, "Hala_B"
 chnset gk_HalaC_VolOut, "Hala_C"

endin


/*                                                                              */
/********************************************************************************/
/********************************************************************************/


/********************************************************************************/
/********************************************************************************/
/*                                                                              */
/*                              INTERNAL RECORD                                 */


instr Record

 puts "Recording to", 1
 itim     date
 Stim     dates     itim
 Syear    strsub    Stim, 20, 24
 Smonth   strsub    Stim, 4, 7
 Sday     strsub    Stim, 8, 10
 iday     strtod    Sday
 Shor     strsub    Stim, 11, 13
 Smin     strsub    Stim, 14, 16
 Ssec     strsub    Stim, 17, 19
 S_outfile sprintf "%s%s%02d__%s_%s_%s.wav", Syear, Smonth, iday, Shor,Smin, Ssec
 puts S_outfile, 1
 fout S_outfile, 18, ga_Mic, ga_Del1_out, ga_Del2_out, ga_Harm1_out, ga_Harm2_out,
  ga_RevA_out, ga_RevB_out, ga_Filt_out, ga_HalaA_out, ga_HalaB_out, ga_HalaC_out

endin

/*                                                                              */
/********************************************************************************/
/********************************************************************************/


</CsInstruments>
<CsScore>









</CsScore>
</CsoundSynthesizer>
```