```
<Cabbage>
form caption("GrainOnFly") size(490, 460), guiMode("queue"), pluginId("def1"), colour(40, 40, 40)
button   bounds( 25, 20, 90, 45), latched(0), channel("Rec"), value(0), shape("square"), colour:0("red"), colour:1("yellow"), text("Rec")
button   bounds(160, 20, 90, 45), channel("PlayOnce"), value(0), shape("square"), colour:0("green"), colour:1("green"), text("Play once")
combobox bounds(295, 20, 90, 45), channel("TabSize"), items("131762", "263524", "527048"), align(centre), colour(orange) ;,value(2)
label    bounds(295, 66, 90, 15), text("Tab Size")

vmeter   bounds(385,  90,  10, 360), channel("RecCount"),  value(0), outlineColour(0, 0, 0), rotate(1.5708, 0, 0), overlayColour(0, 0, 0), meterColour:0(255, 0, 0), meterColour:1(255, 255, 0),
meterColour:2(0, 255, 0), outlineThickness(1)
label    bounds(150, 100, 110,  15), text("Secs. Counter"), align(centre), channel("SecsCounter") ;, channelType("string")
hslider  bounds( 20,  80, 368,  90), channel("StartNdx"),  value(0),                        range(0, 1, .1)
label    bounds(160, 135,  90,  15), text("Pointer"),      align(centre), channel("Pointer")

rslider  bounds(  0, 160,  90,  90), channel("GrDensity"), value(0), text("Density"),    range(1, 60, 20)
rslider  bounds( 80, 160,  90,  90), channel("DevGrain"),  value(0), text("Deviation"),  range(0, 1, .5)
rslider  bounds(160, 160,  90,  90), channel("GrainDur"),  value(0), text("Duration"),   range(.05, .6, .18)
rslider  bounds(240, 160,  90,  90), channel("GrCpsVar"),  value(0), text("Freq. Band"), range(0, .25, .01)
rslider  bounds(320, 160,  90,  90), channel("PanRand"),   value(0), text("Panning"),    range(0, .5, .1)

rslider  bounds(  0, 260,  90,  90), channel("GlAttack"),   value(0), text("Global Att."),    range(.1, 5, 1)
rslider  bounds( 80, 260,  90,  90), channel("GlRelease"),  value(0), text("Global Rel."),    range(.1, 5, 3)

vmeter   bounds(430, 20,  30, 210), channel("vMeter1"), value(0), overlayColour(70, 53, 53, 255), meterColour:0("red"), meterColour:1(0, 103, 171, 255), meterColour:2(23, 0, 123, 255),
outlineThickness(2) ;meterColour:0(0, 255, 0, 255) ;sostituito con "red"
label    bounds(425, 235, 40,  13), text("Mic In"), align(centre), channel("notUsed1")

label    bounds(370, 350, 120, 12), text("StrAppArtS - 2021"), align(centre), channel("StrAppArtS")
keyboard bounds(  5, 365, 480, 90), value(36), middleC(4)
</Cabbage>
<CsoundSynthesizer>
<CsOptions>
-n -d -m128 -+rtmidi=NULL -M0 -dm0 --midi-key-cps=4 --midi-velocity-amp=5
</CsOptions>
<CsInstruments>
;Plugin da esportare come effetto!
;sr     =       44100
ksmps   =       16
nchnls  =       2
0dbfs   =       1

;chnset 1, "InChan"
chnset 0, "vMeter1"
chnset 0, "vMeter2"
gkvuTime    init 0
gkRecOff    init 0
gkPitchKey  init 1
giRecXFade  init .05
giRecTime   init 0
gkRecTime   init 0
giTabLen    init 131762
giTabLenSec init giTabLen/sr

maxalloc 2,1 ; allow only one instance of the recording instrument at a time!
massign 0, 5 ;'5' invia a strumento 'grain', '4' invierebbe a strumento 'loop'

giTabLen    = 131072 ;'fog' vuole lunghezza tab = 2^x!!!
giTabLenSec = giTabLen/sr ;table duration in seconds
;giBuffer    ftgen  0, 0, giTabLenSec*sr, 2, 0; table for audio data storage
giBuffer    ftgen  0, 0, giTabLen, 2, 0; table for audio data storage

;=======
instr 1 ; Trigger record or playback
;=======
   gkInChan        chnget  "InChan"
gkDen chnget "GrDensity"
gkGrDur chnget "GrainDur"
gkCpsVar chnget "GrCpsVar"
gkStartNdx chnget "StartNdx"
gkPanRand chnget "PanRand"
gkDevGrain chnget "DevGrain"
gkGlAtt chnget "GlAttack"
gkGlRel chnget "GlRelease"
kTabSize chnget "TabSize" ;emette '1', '2', '3'
kSizeTrig changed kTabSize

if kSizeTrig > .5 then        ;SCELTA DELLA LUNGHEZZA DELLA TABELLA DI REGISTRAZIONE
reinit resetComboBox
resetComboBox:
gkvuTime = 0                    ;azzera il visualizzatore
cabbageSetValue "RecCount", gkvuTime, 1 ;invia il valore '0' al visualizzatore
if i(kTabSize) == 1  then
cabbageSetValue "StartNdx", .1, 1
giTabLen    = 131072
giTabLenSec = giTabLen/sr ;table duration in seconds
ftfree giBuffer, 0
;giBuffer    ftgen  0, 0, giTabLenSec*sr, 2, 0; table for audio data storage
giBuffer    ftgen  0, 0, giTabLen, 2, 0; table for audio data storage
elseif i(kTabSize) == 2 then
cabbageSetValue "StartNdx", .05, 1
giTabLen    = 263524
giTabLenSec = giTabLen/sr ;table duration in seconds
ftfree giBuffer, 0
;giBuffer    ftgen  0, 0, giTabLenSec*sr, 2, 0; table for audio data storage
giBuffer    ftgen  0, 0, giTabLen, 2, 0; table for audio data storage
elseif i(kTabSize) == 3 then
cabbageSetValue "StartNdx", .025, 1
giTabLen    = 527048
giTabLenSec = giTabLen/sr ;table duration in seconds
ftfree giBuffer, 0
;giBuffer    ftgen  0, 0, giTabLenSec*sr, 2, 0; table for audio data storage
giBuffer    ftgen  0, 0, giTabLen, 2, 0; table for audio data storage
endif
rireturn
endif
```

```
kRec chnget "Rec"
kPlay chnget "PlayOnce"
kRecOn trigger kRec, .5, 0 ;emette '1' quando il pulsante viene premuto
gkRecOff trigger kRec, .5, 1 ;emette '1' quando il pulsante viene lasciato
kPlayTrig changed kPlay
gkModulation    ctrl7   1, 1, 0, .04
gkModulation    portk   gkModulation, .05
gkMidiModul     lfo     gkModulation, 4




  if kRecOn==1 then             ; se premi il bottone di Rec
event   "i", 2, 0, giTabLenSec   ; activate recording instrument
  elseif gkRecOff==1 then       ; se lasci il bottone di Rec
turnoff2 2, 0, giRecXFade       ; deactivate recording instrument
reinit reset
reset:
iRecTime = i(gkRecTime)         ; cronometra e memorizza il tempo trascorso
iPlaybackTime = iRecTime+giRecXFade ;per il playback aggiunge al tempo di registrazione il tempo di release di 'turnoff2'
rireturn
  endif


  if kPlayTrig==1 then          ; se premi il bottone di playback
event   "i", 3, 0, iPlaybackTime  ; activate playback instrument
  endif
;printk .2, gkRecTime
endin

;=======
instr 2 ; record to buffer
;=======
gkRecTime     timeinsts   ;cronometra il tempo di registrazione
gkvuTime = gkRecTime/(ftlen(giBuffer)/sr) ;tempo massimo di registrazione nella tabella riscalato a '1'.
cabbageSetValue "RecCount", gkvuTime, metro(10) ;valori per visualizzatore
cabbageSet metro(10), "SecsCounter", sprintfk("text(\"Secs.: %.2f\")", timeinsts())

  if gkRecOff == 1 then
  reinit reset
  reset:
  giRecTime = i(gkRecTime)
  print giRecTime
  rireturn
  endif


giRecTime = i(gkRecTime) ;durata del campione registrato

kRecEnv    expsegr .0001, giRecXFade, 1, giRecXFade, .0001
ain        inch     1           ; read audio from live input channel 1
andx       line     0, p3, ftlen(giBuffer); create an index for writing to table
           tablew   ain*kRecEnv,andx,giBuffer ; write audio to function table
endin

;=======
instr 3 ; playback once from buffer
;=======
aNdx line 0, p3, p3*sr  ;create an index for reading from table
aRead      table    aNdx, giBuffer  ; read audio to audio storage table
outs       aRead, aRead    ; send audio to output
endin

;=======
instr 4 ;si attiva con massign 0,4 (disattivo)
;=======
iMidiAmp ampmidi 1
print giRecTime
kcpsmidi    cpsmidib 2

kLoopEnv    expsegr .0001, giRecXFade, iMidiAmp, giRecXFade, .0001
gkPitchKey     = kcpsmidi/cpsmidinn(60)                         ; DERIVE RATIO BASED ON NOTE NUMBER 60 AS THE POINT OF UNISON (IE. RATIO=1)
;asig1[, asig2] flooper kamp,     kpitch,              istart,     idur,                 ifad,       ifn
aRead          flooper kLoopEnv, gkPitchKey+gkMidiModul, giRecXFade, giRecTime-(giRecXFade*2), giRecXFade, giBuffer
outs       aRead, aRead
endin

;=======
instr 5 ;si attiva con massign 0,5 (attivo)
;=======
iMidiAmp ampmidi 1
giRecTime = i(gkRecTime)
kcpsmidi    cpsmidib 2
gkPitchKey     = kcpsmidi/cpsmidinn(60)                         ; DERIVE RATIO BASED ON NOTE NUMBER 60 AS THE POINT OF UNISON (IE. RATIO=1)
kGTrig gausstrig  1, gkDen, gkDevGrain
kCpsVar rand gkCpsVar, 0, 0, 1
kRandPan rand gkPanRand, 0, 0, .5
;kGlobalEnv expsegr .001, i(gkGlAtt), iMidiAmp, i(gkGlRel), .001
kGlobalEnv envlpxr iMidiAmp, i(gkGlAtt), i(gkGlRel), 41, 1, .001 ;inviluppo globale, '41' è il numero di tabella letta per l'attacco del suono
schedkwhen kGTrig, 0, 0, 6, 0, gkGrDur, gkPitchKey+gkMidiModul, kCpsVar, gkStartNdx, kGlobalEnv, kRandPan ;attiva i grani di instr 6
endin

;=======
instr 6 ;one grain
;=======
iGrDur = p3
iKey = p4
iCpsVar = p5
iPhs = p6
iGlEnv = p7
iRandPan = p8
kGEnvndx line 0, iGrDur, 1
kGEnv table kGEnvndx, 56, 1 ;inviluppo del grano
```

```
aGrain poscil kGEnv*iGlEnv, sr/giTabLen*iKey*iCpsVar, giBuffer, iPhs
aL, aR pan2 aGrain, iRandPan
outs aL, aR
endin

;========
instr 10 ;DECIBEL LEVEL METER
;========
;from: http://forum.cabbageaudio.com/t/help-how-to-create-a-decibel-level-meter/298/4
    aInL        inch            1
    krmsL       rms                         aInL                                                    ; scan rms
    kdbL        =                           dbfsamp(krmsL)                                  ; convert to db
    cabbageSetValue  "vMeter1", (kdbL+90)/90, metro(10)
endin
</CsInstruments>
<CsScore>
;f 55 0 1024 19 .5 .5 270 .5 ;fog function
f 41 0 129 -7 0 128 1 ;it concerns 'envlpxr', it is the rise shape of envelope.
f 56 0 1024 20 2 ;Hanning window
i 1 0 [3600*24*7] ;Start recording - playback.
i 10 0 [3600*24*7] ;Db level meter
</CsScore>
</CsoundSynthesizer>
```