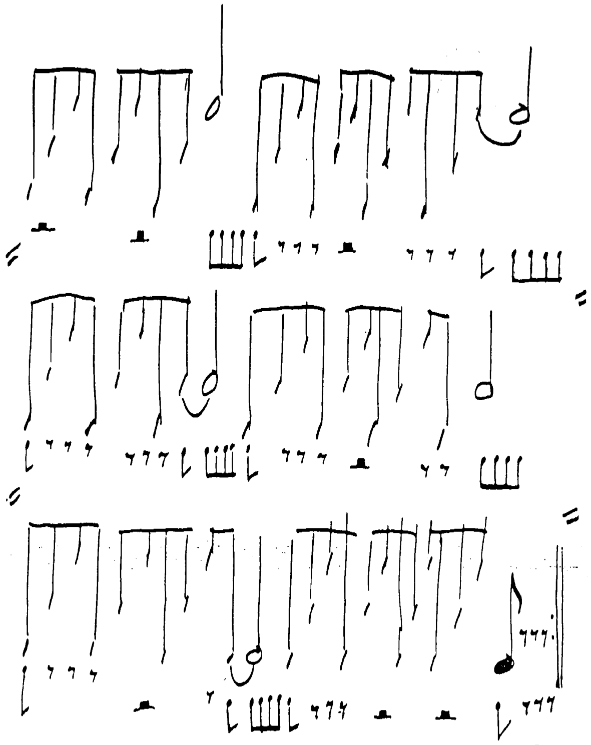
Eine Kleine ~~Gamelan~~ Computer Music

Score by Daniel Goode
Live interactive computer realization by Larry Polansky

HMSL Turnkey Application
Version: December 22, 1995
Shareware

Eine Kleine Gamelan Music

by Daniel Goode computer



© D. GOODE

…………… *for instruments of any tunings and any type* ……………

Construct 3-note melodies of a step followed by a skip, *or* a skip followed by a step in any scale, mode, key, in any tuning, either upward or downward (but not both in the same melody).

Apply these melodies to the motive above, which gives the rhythm and order of the three tones. Play the same melody for *at least two* times through the motive. You can also play the motive mirror-fashion, from the highest to the lowest note and so on.

KEEP the same tone(s), the same variant for at least two repeats before changing or resting.

ENDING: When all instruments are playing the rhythmic variant (eighths in place of long notes), that becomes the last time.

VARIANTS:

–Accent either the lowest or highest tones or both.

–Omit all but the accented tones.

–Play only the long notes and fill them in with eighths including an eighth on the next beat after.* The last beat, as is.

–A sustaining instrument may play a long tone in place of the written eighths, and rest for the long notes of the motive.

–Gongs or gong-like sounds and other percussion instruments can be added to the long notes.

*see rhythmic underlay in score above

Eine Kleine ~~Gamelan~~ Computer Music

Score by Daniel Goode
Live interactive computer realization by Larry Polansky

larry polansky and daniel goode

frog peak music (a composers' collective)
box 1052
lebanon nh 03766
ph/fax: 603-448-8837
email: frogpeak@sover.net
http://www.sover.net/~frogpeak/

larry.polansky@dartmouth.edu
http://music.dartmouth.edu/~larry/polansky.html
ph: 603-646-2139

Eine Kleine ~~Gamelan~~ Computer Music is a shareware application. Distribute it as you like. If you have some fun with it, send us some CD's, stamps, comix, tapes, or something cool.

General Instructions

EK is a turnkeyed application written in HMSL. Double-click on it and go.

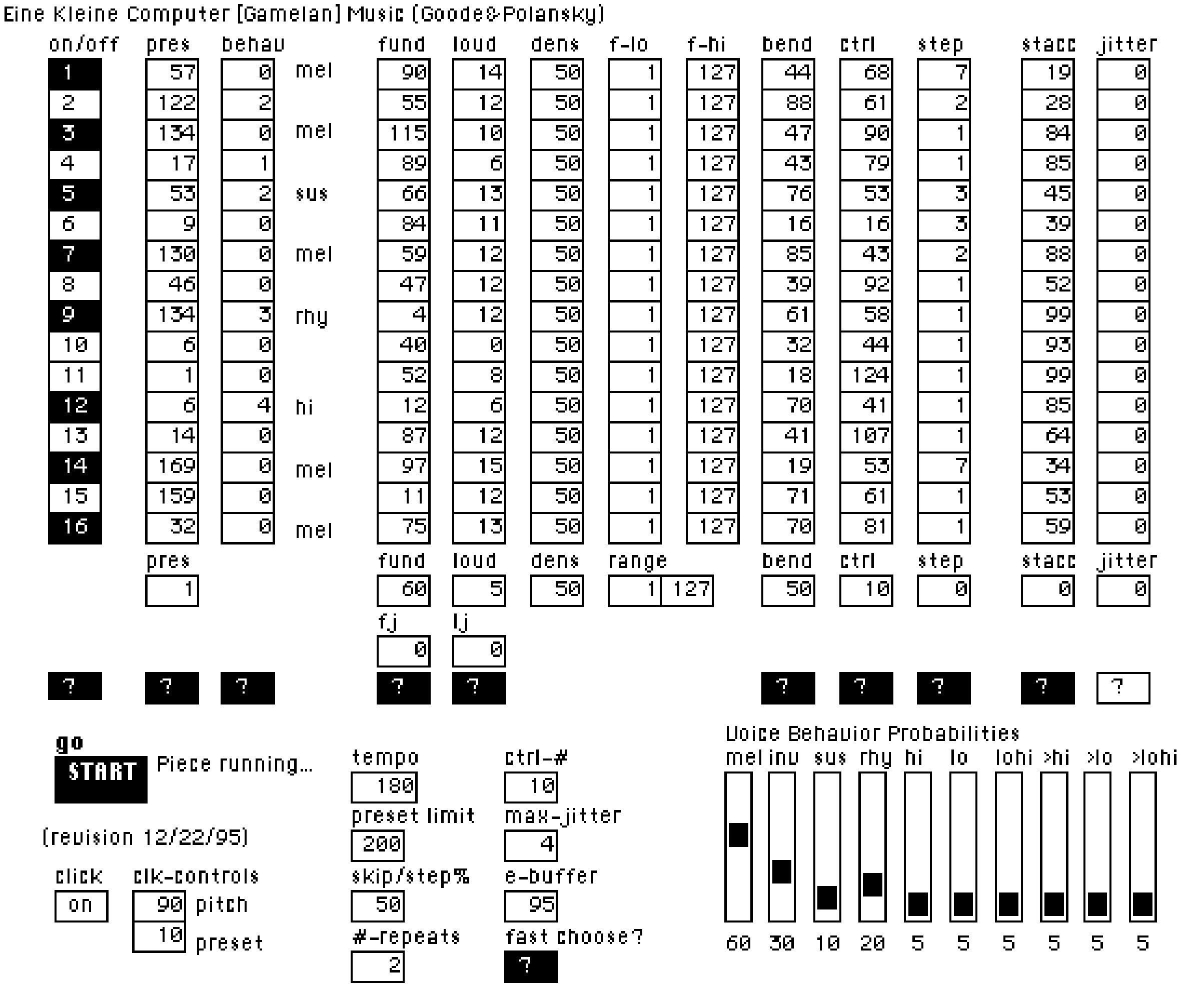
> **IMPORTANT: Get rid of OMS, Free Midi, etc. from your system folder. *EK* uses it's own MIDI driver. Best is to simply turn all extensions off (in Extensions Manager), and use the simplest setting on your MIDI interface (1 Mhz. on a Studio 3-5 will work).*EK* will not send MIDI if there are other MIDI drivers in the system folder. Also, *EK* is meant for a full-page monitor. On a smaller display, some of the screen might not show up.**

What to do

- Connect up to 16 channels of MIDI synth to your Mac
- Get rid of OMS, FreeMIDI, from your system folder (turn all extensions off, probably restart).
- Set your MIDI interface to dumbest mode possible, 1Mhz.
- After double clicking on the *EK* application, click on GO (the piece will start running).

- The piece will run fine, and pretty much forever, continually making changes as it goes. If you want to interact with it, you need to know what all the boxes do (they do a lot!). The rest of this document describes them.
- To quit the piece, you must click on the close box at the top left ("Quit" in the menu is disabled in this application).

The *EK* Screen



Eine Kleine Computer [Gamelan] Music (Goode&Polansky)

on/off	pres	behav		fund	loud	dens	f-lo	f-hi	bend	ctrl	step	stacc	jitter
1	57	0	mel	90	14	50	1	127	44	68	7	19	0
2	122	2		55	12	50	1	127	88	61	2	28	0
3	134	0	mel	115	10	50	1	127	47	90	1	84	0
4	17	1		89	6	50	1	127	43	79	1	85	0
5	53	2	sus	66	13	50	1	127	76	53	3	45	0
6	9	0		84	11	50	1	127	16	16	3	39	0
7	130	0	mel	59	12	50	1	127	85	43	2	88	0
8	46	0		47	12	50	1	127	39	92	1	52	0
9	134	3	rhy	4	12	50	1	127	61	58	1	99	0
10	6	0		40	0	50	1	127	32	44	1	93	0
11	1	0		52	8	50	1	127	18	124	1	99	0
12	6	4	hi	12	6	50	1	127	70	41	1	85	0
13	14	0		87	12	50	1	127	41	107	1	64	0
14	169	0	mel	97	15	50	1	127	19	53	7	34	0
15	159	0		11	12	50	1	127	71	61	1	53	0
16	32	0	mel	75	13	50	1	127	70	81	1	59	0

pres	fund	loud	dens	range		bend	ctrl	step	stacc	jitter
1	60	5	50	1	127	50	10	0	0	0

fj	lj
0	0

Piece running...

(revision 12/22/95)

click: on

clk-controls: 90 pitch, 10 preset

tempo: 180

preset limit: 200

skip/step%: 50

#-repeats: 2

ctrl-#: 10

max-jitter: 4

e-buffer: 95

fast choose? ?

Voice Behavior Probabilities

mel	inv	sus	rhy	hi	lo	lohi	>hi	>lo	>lohi
60	30	10	20	5	5	5	5	5	5

About the *EK* Program

This computer program is a realization of Daniel Goode's 1980 score, which you should consult for the musical ideas referred to below. The program contains some options that are not explicitly "allowed" in the original, but you don't have to use these if you don't want to.

Every time the program completes a certain number of repeats of the one-page score, it chooses a new set of values for all the voices. You'll see the values change on the screen. You can, in some cases, specify the ranges from which those values will be chosen. Although the score specifies that you can't change a musical idea until you've repeated the tune at least twice, this program allows you to set the *#-repeats* before choosing a new set of values. The default is 2.

About the *EK* Screen

The *rows* are the 16 midi channels (performers, voices). The *columns* are the different parameters for each voice, controllable by the user. The program selects new values for each voice in each parameter for each new repeat, if that parameter (column) is enabled for choosing.

Under each column (except the voice *on* and *behav*) is a single box which allows you to globally force all the voices in that parameter to one value. This can be useful for specific musical effects, such as changing all the fundamentals, or loudnesses together.

The boxes at the bottom center of the screen (*tempo*, *preset-limit*, *skip/step%*, *#-repeats*, *ctrl-#*, *max-jitter*, *e-buffer*, *fast Choose?*) are for controlling more global aspects of the piece. They will be described below in detail.

The 10 sliders (*Voice Behavior Probabilities*) are for controlling the likelihood that a voice will choose one of 10 different musical behaviors for a repeat. These are described below.

The Columns

Below each column of 16 voices (except for *on* and *behav*) are two boxes, one with a number in it, one with a question mark. These affect the columns globally, that is, they affect all 16 voices at once. They'll be described below. The individual cells in the column each affect one voice.

- *on/off*

This turns a voice on or off, and displays whether or not a voice has been turned on or off for a given repeat. The probability of a voice turning on or off is controlled by the *dens* (density) column, and goes from 0 (no possibility) to 100 (turns on every time). Different voices are different MIDI channels.

- *pre*

  This selects a preset or program number for a voice, and displays the program number chosen by the program. You can also set a limit for the highest preset number the piece will choose (when it is doing the choosing) in the box in the bottom center called *preset limit*.

- *behav*

  You cannot alter anything in this column, it's for display only. (You can change the numbers in the column, but they don't really do anything: the text, telling you what behavior is in effect, won't change). This column tells you what "behavior" the voice is in for a given repeat. The behaviors are, according to Goode's score:

  — (0) play the melody (*mel*)
  — (1) invert the melody (*inv*)
  — (2) sustain one note through the 8th notes of the melody (*sus*)
  — (3) play equal eighth notes on one pitch through the long notes (*rhythm*)
  — (4) play only the high notes of the melody (*hi*)
  — (5) play only the low notes of the melody (*lo*)
  — (6) play the low and high notes of the melody (*lohi*)
  — (7) accent the high notes of the melody (>*hi*)
  — (8) accent the low notes of the melody (>*lo*)
  — (9) accent the low and high notes of the melody (>*lohi*)

  The probability of picking one of these 10 different behaviors is controlled by the *Voice Behavior Probabilities* sliders on the bottom right of the screen. The probabilities for any given behavior range from 0->100. Next to the individual boxes in the *behav* column, the name of the behavior currently in use is displayed.

- *fund*

This column displays and allows you to change the fundamental pitch for each voice, to which the step and skip are added (or subtracted in the case of the inverting behavior). These numbers refer to MIDI note numbers (60 is middle C). If the voice selects its own fundamental for a repeat, the value will be in the range specified by the two columns *f-lo* and *f-hi*. This fundamental will also be the note used by the sustain and rhythm behaviors.

- *loud*

  The numbers in this column range from 0-20, and are the user- or program-selected scalars for the loudness of each voice. To make a voice louder, raise the number, to make it quieter, lower it.

- *dens*

  This column is the probability of a voice being turned on for the next repeat. It ranges from 0-100. It is not chosen by the program.

- *f-lo*

  This column is the lower limit for the choice of a new fundamental for the voice. It is not chosen by the program. Units are MIDI note numbers.

- *f-hi*

  This column is the higher limit for the choice of a new fundamental for the voice. It is not chosen by the program. Units are MIDI note numbers.

- *bend*

  Numbers in this column range from 0-100, and control and display the amount of pitch bend sent to a given voice for each note. The numbers control the percent of the total pitch bend range, and the actual pitch bend values sent are randomly chosen from between 0 and +- the range. That is, if *bend* is set at 10, then pitch bend values will be between +10% and -10% the total pitch bend range. For a simple detuning, set your MIDI synthesizer to have a pitch bend range of 1 semitone.

- *crtl*

  The numbers in this column range from 0-127, and control the amount of some MIDI controller that will be sent for a given voice for each note. The actual pitch controller values sent are randomly chosen from between 0 and the number specified in the box. By changing the value in the box called *ctrl-#* on the bottom center of the screen, the MIDI controller # to which the values are sent is changed. 10, the default, is the usual controller number for panning. By setting different MIDI channels to respond differently to the same controller #, a wide variety of effects can be created using this feature.

- *step*
  Picks and displays the step size in semitones for the melody. The skip size is recomputed every time, as something between 2 and 4 times the step size. The skip size is not displayed.
- *stacc*
  Picks and displays a staccato value for the voice, between 0-100% of the duration of the notes. This will affect different MIDI programs in different ways. Basically, it determines when the noteoff will be sent (in terms of the % of the total duration).
- *jitter*
  My five year old daughter Anna calls this the “crazy” place. Values for jitter determine the amount of duration error around the pulse for a given voice. Values below 10 will make the piece just a little bit “sloppy” (or lifelike perhaps), greater values will produce odd effects. The voices come back in sync at the beginning of each repeat. Values for jitter are percentages of the total duration of the current pulse. You can set the maximum jitter value that will be chosen for the voices on each new repeat in the box at the bottom center called *max-jitter*.

Global Column Parameters (under each column)

- *numeric parameters*
  Below each column is a box with a number in it. Changing that number will force all the voices to that value in that parameter. There is no global changer for the behavior or on/off column. If you want all voices to go on and stay on for the next repeat, turn them all on and turn off the choose enable (or set all their *dens* values to 100, which will turn them all on for the next repeat).
- *choosing on/off*
  By blackening and un-blackening the question marks, the possibility of choosing new values in that particular parameter in the next repeat are

enabled and disabled. If you want the program to choose new values for a given parameter, click on this box and make it black. If you don't, click on it and un-blacken it.

- *fj* , *lj,* and *cj*

  These are individual "jitter" factors for the global changers for fundamental, loudness, and control. They specify a value which is the range *around* the value specified in the global changer for that parameter. In other words, if *fj* is set to 2, then when you move the global fundamental to 60, values will be between 58-62. It's a way of dragging all 16 channels of these three parameters into a range, not just a constant value. Setting these boxes to 0 means that all values will move to the values specified in the global parameter changer.

Global Parameters (bottom center of the screen)

- *tempo*

  This selects the number of eighth notes per minute.
- *preset-limit*

  This number is the highest preset that the program will choose for a given voice. For example, if your synth only has 30 presets, set it to 30.
- *skip/step%*

  Daniel Goode's score says that when the melody is played, it can be skip followed by step, or step followed by skip. This number sets the likelihood of picking one of those behaviors over another when *mel*, *inv*, *>lo*, *>hi*, or *>lohi* are chosen. The higher the number, the more likely a voice is to use a melody which is a step followed by skip.
- *#-repeats*

  This number is the number of times the one page score will be played before selecting new values for all parameters enabled. Daniel Goode's score says that performers should remain on a set of choices for at least two times through, but in this program, you can select 1 (good for trying things out).
- *ctrl-#*

This selects the MIDI controller number to which the *ctrl* values are sent for each note. The default is 10, which is often pan.

- *max-jitter*

  This number is the maximum jitter value that will be selected for voices for each repeat. Keep it low (below 10) for a kind of "human" effect. Higher than that, things get strange. For completely accurrate rhythms, set it to zero.

- *e-buffer*

  The event-buffer is specified in units of the tempo. For example, if the tempo is 180 (8th notes per minute) and the event-buffer is 90, it is a half of a second long. This means that there will be a half second lag between things you do on the screen (change parameters, etc.) and MIDI output results. The event-buffer allows the computer extra time to compute, so the output won't be affected by time deformations. Generally it can be left alone, but it might be useful if, for example, you're running *EK* on a really slow Macintosh, or the machine is getting bogged down and not keeping up with the tempo. With a small event-buffer, delay problems will be most noticeable on the very first note of a new set of repeats, when the program is making a lot of calculations (especially if all densities are high, and a lot of parameters are enabled for choosing). That note might be truncated, late, or just sound a bit funny You can use the e-buffer to fix this (also, enable *fast-choose*, below). The e-buffer number sets the length of the HMSL event buffer. The longer the buffer, the more accurately things will compute (that is, there won't be time deformations as the computer "thinks") but the program will always be that amount of time behind "real-time" (that shouldn't affect much in this piece). Musical uses of the event-buffer include being able to "preset" things on the screen a bit before they actually occur, synchronizing changes for the beginning of new repeats.

- *fast-choose?*

  Turning this on tells the program to ignore, for the purposes of choosing new values for parameters, any voices that are not currently on, or turned on in the new repeat. In other words, this is a way of speeding up the choosing process, by not choosing values for voices that are not playing If you turn those voices on in the middle of a repeat, they'll have the values from the previous repeat, or the last time they were chosen. This is simply another way of optimizing for speed in case the machine gets bogged down on the first note. In most cases, it won't be necessary. However, it is also a way of stabilizing given parametric values until you turn them on again, and can be used musically.

- *click*

  The click function turns MIDI channel 16 into a quarter note click track, and excluding it from the normal behavior of the piece. Turning that box on will start the click on the next repeat. Turning it off will mute the click, and allow channel 16 to be used in the usual way. When the click is on, none of the other cells affect its behavior, it's pitch and preset are controlled by the two *clk-controls* boxes. You cannot change its loudness on the screen: do that on your synth. The idea is that one MIDI channel can be dedicated over a separate audio channel (whether by a single synth on that channel, or by using the multiple audio outputs of a synthesizer) to a click track which can aid live performers playing with the piece.
- *clk-controls*

  These two boxes control the pitch (top) and preset for the click track. They have no effect on the rest of the grids on the screen.

Voice Behavior Probabilities

The faders at the bottom right of the screen are a set of probability histograms for the various types of behavior mentioned in the score and described above: melody, sustain through the melody, playing the rhythms on the held notes, invert the melody, accent the low, high, or high and low notes, just play the low, high, or low and high notes.

Acknowledgments

Eine Kleine ~~Gamelan~~ Computer Music was developed as part of a residency at P.A.S.S in NYC by Daniel and Larry. Thanks to the wonderful and talented Alex Noyes for making that residency so great. Thanks also to Nick Didkovsky for his support, and debugging help. My daughter, Anna Diamond Polansky (age 5) beta tested the piece every night for a few weeks while brushing her teeth. HMSL is written by Phil Burk, Larry Polansky and David Rosenboom, and is distributed by Frog Peak Music (A Composers' Collective).

Daniel Goode's *Eine Kleine Gamelan Music*, written in 1980, was published in Ear Magazine, 8:4, Sept. - November, 1983 (p. 7). The original piece has been performed a great many times by Goode, members of Gamelan Son of Lion, and others.

larry polansky
lebanon, new hampshire
(Begun: Sept. 1995)
December 27, 1995