

LAB NOTES:

PINK TUNES

By: John S. Simonton, Jr.

As we begin this month's journey into the bizarre I should warn you that I'm operating in a somewhat altered state of consciousness.

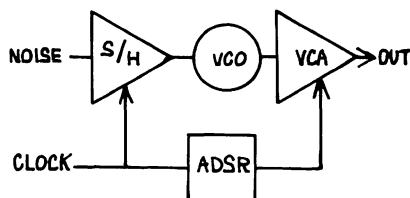
Oh, not from chemicals or nature's own, none of that. There's just some very nice color graphics going on the Apple II and the background music is a slightly oriental feeling 4 part harmony being composed by a P4700/J. It's really a most unique environment.

Wait. Composed by the synthesizer? Surely not, surely just something pre-recorded and played back.

Well, I suppose that I wouldn't attack someone who asserted that I composed the piece. I'd be flattered, but it wouldn't be entirely correct. I knew before the tune started what sort of texture (for lack of a better term) it would have. But I have no idea what exactly is coming next.

And that, in case you hadn't already guessed, is what we want to talk about this time. Computer programs that compose music.

Let's start at a very elementary level. Probably you've seen or connected synthesizer patches that look something like this:



It's a relatively common configuration in which, at regular intervals, the instantaneous value of the noise source is captured by the S/H and the resulting voltage used to set the pitch of the VCO. The ADSR and VCA give us some knobs to twiddle and control dynamics, but otherwise are just window dressing.

If you've done one of these, you know that the results are interesting, but certainly not a musical composition in the traditional sense. As a composing device, it's hard to know what the biggest fault is here, but certainly it must be the fact that there are no guarantees that the series of pitches produced are going to be equally tempered intervals (or any known tempering for that matter). In fact,

you can almost guarantee that they won't be; the control voltages applied to the oscillator are completely random.

And therein lies the tale.

I don't believe that anyone is able at this point in human development to concisely explain what makes music "musical", but most folks that have thought about it seem to feel that "good" music (ugh, all the subjective terms) combines both order and disorder. Establish a pattern in the listener's mind... then surprise him; pleasantly, preferably.

Like the "noise music" example above, any compositional program that we come up with today is in some way going to rely on a STOCHASTIC (big word for random) process. If it didn't, we wouldn't be writing the program that wrote the music; we'd be writing the music.

Our task then, is to bring order from disorder (in a very real sense, nothing less than reversing entropy) - but not completely. It isn't easy, but in an elementary form not as difficult as it may sound either because we now have at our disposal that wonder of wonders (which many right-thinking people say is Maxwell's Demon personified):

THE COMPUTER

By simply programming the computer to randomly select only pitches that are part of the equally tempered sequence, we've made a start, but in all honesty not much of one; still there is too much disorder. Low pitches followed as likely as not by very high ones, no identifiable key signature. It's still "noise music".

The quickest way to begin bringing the kind of order that we're looking for is to write a program that uses a random number not as the note, but as a "pointer" which is used to select one of a number of acceptable "candidate" notes from a previously entered table. We're using our intellect to select ahead of time only those notes which we know will harmonize with the rest of the notes which the computer is allowed to select. I've written a few of these kinds of programs. They're a little better than purely random notes, but not much. Still too much disorder.

There are a lot of tricks to bring

rigorous order, like making random substitutions of candidate notes into previously entered melody lines. This kind of thing produces terrific results, but it's not the computer doing most of the composition - you are.

Now comes the April issue of Scientific American and there, in Martin Gardner's consistently enlightening Mathematical Games column, is a piece on computer music. Well, not just computer music - as is usual, Mr. Gardner's mind ranges far and the column covers visual art and computer generated "landscapes" and fractal curves and the place of pink noise in "the meaning of it all". Very heavy. And buried in amongst it all is an algorithm conceived by Richard Voss (of IBM) for turning "white" random numbers "pink".

Don't let this "white" and "pink" business throw you. You're used to white noise and the pink noise that results when you filter it. We can think of the Voss algorithm as a filter for random numbers.

The realization of the Voss algorithm which is used in PINK TUNES (the program listing at the end of this column) can be likened to rolling a set of 5, four sided dice whose faces bear the numbers 0 - 3. We get the random number that we'll use as the pointer to the list of candidate notes by adding together the numbers on the exposed face of each die (I know, a 4 sided die won't have an upper face, that's not the point).

If we consistently rolled all 5 dice, we would still produce too random a number; even through, as any craps shooter can tell you, the probability is that the total of the faces will be somewhere in the middle of the range of possible numbers - just as a pair of six sided die "like" to come up 7.

The trick is not to roll all 5 dice every time, but rather to come up with a scheme that most frequently rolls one or two and infrequently rolls all 5. Since the random number that is produced is always a total of the 5 dice, this produces a series of numbers that most frequently vary only slightly from one another while still permitting periodic large changes.

Voss's scheme (and ours) is to maintain a 5 bit "pinking counter" (our term) which is incremented each time we get ready to generate a new pink number. The new value of the pinking counter is compared to the old and only those "die" which correspond to bits in the counter which have changed are rolled.

The rest of the program is "over-head". As I mentioned in the beginning, PINK TUNES actually generates a 4 part harmony (provided that we supply it with harmonizing notes in the candidate list) and the program must keep track of how long each of the notes in the 4 parts is to play and allow for the updating of the candidate list and recognize a limited number of commands from the computer's keyboard.

The fully documented listing is the best place to go to see how it all works (it's in your best interests to understand it as fully as possible) and specific details and asides are covered in the boxes.

After entering the program and its data base (note part of the program is on page zero, part is on page one and the data base and working registers are on page zero), first save a copy on tape. If something goes crazy, you don't want to have to enter it all again.

Set up the synthesizer and start running the program starting at the hard start location of \$0003. The data that you loaded is for the pentatonic scale composition that I mentioned in the opening paragraph and you should immediately hear the synthesizer producing the composition. It should go without saying that you will undoubtedly have to call the tuning function (control key #1) and tune the oscillators before it makes music.

You have the ability to change the candidate note list while the program is running simply by pressing keys on the keyboard, but bear in mind that the candidate list is 16 notes deep. As you enter a new note, the one that was entered "16 notes ago" disappears from the list. If any of the 16 notes are inharmonic, the program will periodically produce discordant sequences.

With PINK TUNES running, three of the computer's control keys have meaning:

Key 0 "scrambles" the random number generator to produce a new tune. This is really only useful if you are in the cyclic mode (see box).

Holding key 1 provides a tuning function by causing all 4 outputs to produce a triggered middle C.

Touching key #2 initiates a muted shut-down of the synthesizer and branch back to the monitor, allowing changes in the memory locations described in the boxes.

After making changes using the monitor, always start the program running again from the soft start location \$000B.

The program runs very nicely, but is experimental and not intended as a finished product. Skillful polishing should reduce its length by at least 15 - 20% and it would be nice to make changes in timing, etc. 'on the fly' without having to shut down the synthesizer.

At the same time that the program is primarily "just for fun", don't dismiss it as trivial. It definitely produces 4 part harmonies and even those that are not directly useable in a composition can

serve as inspirational lubrication to the gears of creativity. If you're involved in producing commercial jingles, this is a terrific tool.

As you play with the program you will begin to get a feel for how various probabilities affect the composition and you're sure to learn some things about composition that you never knew before.

Finally, a very special thanks to Bob Yannes who sent me a listing of a similar program (PINK FREUD) which generates 4 part canons on a P4700/J. I haven't reviewed this program thoroughly yet, but knowing Bob it's sure to be neat. I'm sure that he wouldn't mind my sharing copies of the listing with anyone who sends a SASE.

















'Til next time, my best to all.








NOTE DURATIONS

Each of the 4 output channels has associated with it its own duration timer and two variables in the computer's memory which determine what characteristics the time values of the notes produced by that channel will have. In the interest of convenience, we'll name these two variables MASK and TIME; or, simply M and T.

We need to think of each of these variables as being composed of a high half-byte (hbb) and a low half-byte (lhb). The hex number \$F3 (an arbitrary example) has an hbb of \$F and an lhb of \$3. This is necessary because the half-bytes determine two separate parameters.

The lower half-bytes of MASK and TIME (M1 and T1 respectively) interact to determine what time values are possible from a given channel. A channel can be restricted so that it produces only 1/16 notes or 1/16 and 1/8 notes or a wide variety of other possibilities as summarized in the table below:

	M1			
	0	1	2	3
T1				
				
				
				

KEY:
 = sixteenth note duration  = whole note duration
 = eighth note duration  = two whole note
 = quarter note duration  = three whole note
 = half note duration

Note that this is a partial table intended only to demonstrate the pattern.

Other combinations of M1 and T1 produce other possible time values. Some combinations not listed will produce undesirable results.

The high-half-bytes of MASK and TIME (Mh and Th) interact to determine the probability that the note being produced by that channel will be dotted (its duration extended by half of its actual value).

In actual practice, it is most convenient to set Mh to \$F and regulate the probability using only Th. The influence of Th on the probabilities of a dotted note is illustrated below:

Th	Probability of dotted note
\$8	one in two
\$4	one in four
\$2	one in eight
\$1	one in sixteen
\$0	zero

EXAMPLE: A channel which has MASK and TIME values of \$F3 and \$11 respectively will be capable of producing 1/16, 1/8, 1/4 and 1/2 notes with a one in sixteen probability of the note being dotted. A channel with M and T of \$F0 and \$01 will produce nothing but 1/16 notes, none of which will be dotted.

The page zero addresses of the MASK and TIME parameters for the four output channels are given below:

	CHANNEL			
	A	B	C	D
MASK	\$8F	\$8E	\$8D	\$8C
TIME	\$8B	\$8A	\$89	\$88

TEMPO

By using the MUS-1 subroutine LOOK to gather data from the AGO keyboard, PINK TUNES follows our

standard protocol of using the key-board encoder clock rate as the system master clock. Analog control of tempo may be provided by varying this clock rate as has been mentioned in previous columns.

PINK also has a variable at zero page location \$A9 which gives gross digital control of tempo. The recommended range of values for this variable are from \$FF (far too fast) to \$F0 (insanely slow).

GLIDE AND TRANPOSE

PINK uses the MUS-1 QuASH drivers (NOTE) and therefore allows for both independent pitch transpositions of any and/or all 4 channels as well as providing a means of enabling or disabling glides.

Though not strictly true, it is most convenient to think of these variables as being divided into high half-byte and low half-byte with the hbb controlling glide (\$8 turns the glide on, \$0 turns it off) and the lhb determining transposition. For example, a channel which has this transposing variable set to \$8C will have its glide turned on and be playing notes an octave higher than the actual note selected by PINK.

Here are the transposing variable addresses:

	CHANNEL			
	A	B	C	D
TRANPOSE	\$CF	\$CE	\$CD	\$CB

CYCLE CONTROL

The variable at zero page location \$D3 controls the number of notes which will be played before the cycle repeats. Changing the contents of this location to \$20 (for instance) will cause 4 bars of eighth notes to be played before the tune repeats. \$40 would produce 8 bars of eighth notes.

Setting the contents of the location to \$00 amounts to enabling a "free run" mode in which the patterns do not repeat (in practical terms).

If you want to get really fancy, you can change program location \$188 from its current value of 85 (STA to the zero page) to EA (a NOP) and the result will be that on successive cycles the time values of notes will not change but the actual notes played will, producing a strong rhythmic tie from cycle to cycle. It also doesn't always work, sometimes a repeating loop will be entered anyway. Other times the duration of a tune will be 2 or more times as long as the actual cycle time.

To change a cyclic tune, touch control key 0.

DE-PINKING

To get some feel for the effect that the Voss pink-ing algorithm has on the composition, you may want to change it slightly. There are a couple of easy ways that this can be done. By changing the current instruction at program location \$11C from \$45 (Exclusive-OR on the zero page) to \$EA (a NOP), you slightly de-pink the note selector, making it somewhat more random. You may have to listen a while before you notice the difference, but there is one.

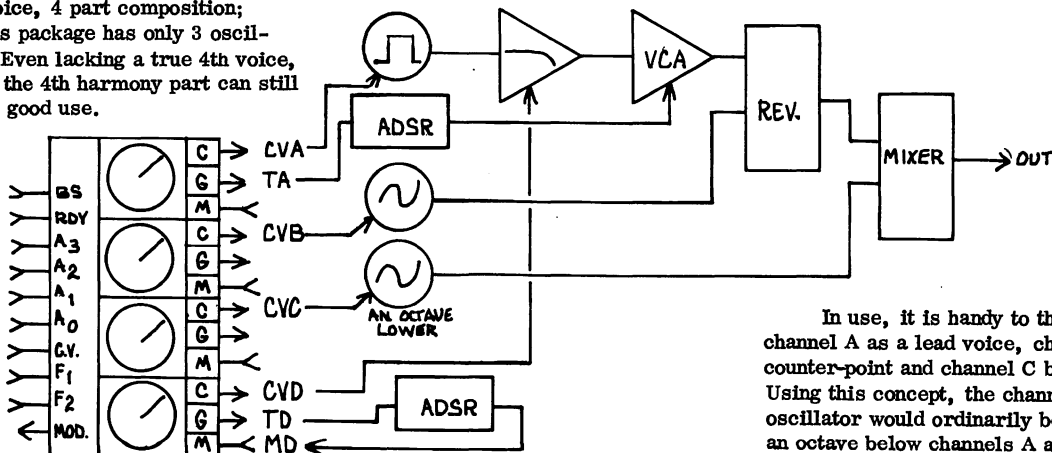
To completely eliminate the Voss algorithm make these substitutions beginning at location \$118; A9 FF EA EA EA EA. This change is equivalent to rolling all 5 of our alleged 4 sided dice each time a new note is selected and will produce changes that even a tone deaf aborigine would recognize.

THE SYNTHESIZER

The module complement of a P4700/J is not large enough to do a true 4 voice, 4 part composition; since this package has only 3 oscillators. Even lacking a true 4th voice, however the 4th harmony part can still be put to good use.

Here is the most universal of the patches used during the development of PINK TUNES:

Note that the 4th harmony part (from channel D of the QuASH) is used to set the center frequency of the VCF.



In use, it is handy to think of channel A as a lead voice, channel B counter-point and channel C bass line. Using this concept, the channel C oscillator would ordinarily be tuned an octave below channels A and B.

THE CANDIDATE NOTES

Selection of the candidate notes that you give PINK TUNES and the order in which they're entered play a big part in the feel of the final composition. As an obvious example, the pseudo-pentatonic scale resulting from entering only accidentals (sharps and flats) tends to produce oriental sounding compositions.

The selection of notes is "pinked" on a compositional (rather than a per-channel) basis, which means that the 4 notes being played at any one time tend to cluster around a relatively short series of entries in the candidate table. The significance of this is that it allows statistical control of changes in key signature. For example, entering the candidate sequence C1, E1, G1, C2, E2, G2, C3, G2, A2, F2, D2,

A1, F1, D1, F1, A1 will produce a composition that periodically changes from the key of C to D minor.

It is important to remember that the candidate table will always contain 16 notes and in order to produce consistent harmonies, all 16 notes must be harmonious. Also remember that notes at the ends of the table (oldest and newest entries) have a lower probability of being played than the notes in the middle.

LOADING THE PROGRAM

NOTE THAT PINK TUNES CONSISTS OF THREE MAJOR SECTIONS: THE MAIN PROGRAM ON PAGE 0 OF MEMORY, SUBROUTINES ON PAGE 1, AND DATA BASE ON PAGE 0.

BEFORE ENTERING ANY PROGRAMMING, MAKE SURE THAT THE MONITOR STACK AND USER'S STACK ARE BOTH SET TO \$FF (SO THAT THE STACK DOES NOT OVER-WRITE PROGRAMMING ON PAGE 1) AND THAT THE STATUS REGISTER IS SET TO \$00 (TO INSURE THAT THE CPU IS WORKING IN THE HEXADECIMAL MODE) USING THESE ENTRY SEQUENCES:

00D-DISP-FF-ENT (SETS MONITOR STACK)
00E-DISP-FF-ENT-00-ENT (USER STACK
AND STATUS REGISTER)

ALL OF THE FOLLOWING PROGRAMMING, DATA BASE, AND INITIALIZATION OF MUS-1 NEED BE DONE ONLY ONCE. THEY WILL SUBSEQUENTLY LOAD TO THE COMPUTER'S MEMORY FROM THE MASTER TAPE THAT YOU WILL GENERATE AT THE END OF THE LOADING PROCESS.

INITIALIZE THE MUS-1 VARIABLES
CTRL (\$0E8) AND ONLY (\$0E9)

008-DISP-00-ENT-20-ENT

ENTER THE DATA BASE LISTED BELOW BEGINNING AT LOCATION \$008 USING THIS ENTRY SEQUENCE:

008-DISP-02-ENT-04-ENT-01-ENT (ETC)

DATA BASE

008:02 04 01 01 F2 F0 F3 F3
5A 5D 5F 62 64 62 5F 5D
5A 58 56 53 51 53 56 58
00 00 00 00 00 00 00
FA

NEXT LOAD THE MAIN PROGRAM:

000-DISP-4C-ENT-C0-ENT-FF-ENT (ETC)

AND THE SUBROUTINES:

100-DISP-8A-ENT-48-ENT-A5-ENT (ETC)

BEFORE TRYING TO RUN THE PROGRAM SAVE IT ON TAPE FROM LOCATION \$0 TO \$190:

0-0-0-0-0-1-A-0-0-1-D-D-TAPE

BEGIN RUNNING THE PROGRAM FROM THE 'HARD START' LOCATION \$3:

003-RUN

AFTER A SHORT (3 SECONDS OR SO) DELAY, THE PROGRAM WILL BEGIN PRODUCING THE COMPOSITION.

THE 'SOFT START' LOCATION IS \$008

```
0010 :*****
0020 :*
0030 :*      PINK TUNES      *
0040 :*
0050 :*      A COMPOSING PROGRAM      *
0060 :*      FOR FOUR PART HARMONIES  *
0070 :*
0080 :*      BY JOHN S SIMONTON, JR  *
0090 :*(C) 1978 PAIRA ELECTRONICS, INC *
0100 :*****
0300 :
0310 :
0320 :FIRST ATTEND TO HOUSEKEEPING--
0360 :
000- 4C C0 FF 0370 BEG JMP BRAK :BREAK VECTOR
003- 20 21 00 0380 STAR JSR INIT :SET UP SYNTH
006- AD 10 08 0390 LDA KBD :INITIALIZE RANDOM
009- 85 D0 0400 STA #TMP+01 :NUMBER GENERATOR
00B- 20 71 01 0410 LOOP JSR SET :INIT PINK TUNES
00E- 20 2B 00 0420 LP0 JSR NOTE :PLAY NOTES READ AGO
0430 :
0440 :CHECK FOR ADDITIONS TO CANDIDATE
0450 :NOTE TABLE
```

```
0470 :
0480 :MAIN LDA #KTBL+00 :ANY KEYS DOWN?
0490 :BEQ OUT1 :NO-CHECK FOR TIME OUT
0500 :CMP #TEMP :YES-A NEW KEY?
0510 :OUT1 STA #TEMP :SAVE FOR NEXT TIME
0520 :BEQ OUT :BRANCH IF SAME KEY
0530 :
0540 :LDX 10 :IF NEW KEY SHIFT
0550 :LP3 LDY #NBUF,X :ALL 16 CANDIDATES
0560 :STA #NBUF,X :DOWN BY ONE
0570 :TYA
0580 :DEX
0590 :BNE LP3 :NOT DONE-LOOP
0600 :
0610 :NOW CHECK FOR CLOCK TIME OUT
0620 :
0630 :OUT LDA #CLK :GET MASTER CLOCK
0640 :BNE TEST :AND IF TIMED OUT
0650 :LDA #TMP0 :SET TO TEMPO VALUE
0660 :STA #CLK :CALL SUB FOR NEW
0670 :JSR ALOC :NOTES (IF NEEDED)
0680 :LDA #LNTH :GET CYCLE STATUS
0690 :STA DISP :SHOW IT AND IF ZERO
```

```

035- F0 00 0700 BEQ TEST :CYCLE IS COMPLETE
037- 06 A8 0710 DEC *LNTH :IF NOT DONE, DORMIT
039- 00 00 0720 BNE TEST :IF NOT ZERO NOWLEAVE
038- 20 71 01 0730 JSR SET :IF ZERO, REINIT
03E- 20 53 01 0732 JSR ALOC :GET FIRST NOTES AND
041- F0 00 0735 BEQ LP0 :BRANCH ALWAYS TO PLAY
043- 20 00 0F 0740 TEST JSR DEC0 :GET A COMMAND
046- 00 00 0750 BNE TST2 :NOT ZERO, NEXT TEST
048- A2 03 0755 LDX 03 :COMMAND 0, NEW TUNE
      0757 :SET POINTER/COUNTER
04A- 20 00 01 0760 TST1 JSR RNDM :GET RANDOM NUMBER
04D- 95 CF 0762 STA *NTMP,X :NEW INITIAL RANDOM
04F- CA 0764 DEX :POINT TO NEXT
050- 00 F8 0766 BNE TST1 :NOT DONE - LOOP
052- F0 B7 0770 BEQ LOOP :BRANCH ALWAYS
054- C9 01 0780 TST2 CMP 01 :COMMAND 1, TUNEING
056- 00 00 0790 BNE TST4 :NOT 1, TEST NEXT
058- A2 04 0800 LDX 04 :4 OUTPUT BUFFERS
05A- A9 5C 0810 LDA 5C :PUT MIDDLE C IN ALL
05C- 90 D8 00 0820 TST3 STA NT08,X :OUTPUT BUFFERS
05F- CA 0830 DEX
060- 00 FA 0840 BNE TST3 :NOT DONE-LOOP
062- F0 AA 0850 BEQ LP0 :BRANCH ALWAYS
064- C9 02 0860 TST4 CMP 02 :COMMAND 2, STOP
066- 00 A6 0870 BNE LP0 :NO COMMAND - LOOP
068- 20 71 01 0880 JSR SET :CALL TO ZERO OUT-BUFFS
06B- 20 2B 00 0890 JSR NOTE :THEN MUTE SYNTHESIZER
06E- 00 0900 BRK :AND RETURN TO PIEBUG

```

SUBROUTINES

```

0220 : RANDOM NUMBER GENERATOR
0230 :
0231 : ESSENTIALLY A 22 BIT LONG SHIFT
0232 : REGISTER WITH EX-OR TAPS AT
0233 : STAGES 22 AND 21 FED BACK TO
0234 : INPUT.
0235 :
100- 0A 0240 RNDM TXA :SAVE X
101- 48 0250 PHA
102- A5 A5 0260 LDA *NOIS+01 :LAST BYTE S/R
104- 0A 0270 ASL :ALIGN BITS 22 &
105- 45 A5 0280 EOR *NOIS+01 :21 AND DO EX-OR
107- 0A 0290 ASL :THEN SHIFT RE-
108- 0A 0300 ASL :SULT TO CARRY
109- 0A 0310 ASL
10A- A2 03 0320 LDX 03 :SET UP PNT/CNT
10C- 36 A4 0330 LP1 ROL *NOIS,X :AND SHIFT 3 BYTE
10E- CA 0340 DEX :SHIFT REGISTER
10F- 00 FB 0350 BNE LP1 :BY ONE BIT LEFT
111- 68 0360 PLA :WHEN DONE RE-
112- AA 0370 TAX :STORE X REG.
113- A5 A7 0380 LDA *NOIS+03 :AND LEAVE WITH
115- 60 0390 RTS :WITH NO. IN ACC.
0400 :
0410 : NEW NOTE
0411 :
0412 : TAKES CARE OF PICKING PINK NOTE
0413 : FROM CANDIDATE NOTE TABLE AND
0414 : CALCULATES AND UPDATES NOTE TIMERS
0415 : NOTE THAT Y POINTS TO CHANNEL FOR
0416 : UPDATE
0420 :
116- A2 05 0430 NMNT LDX 05 :SET UP PNT/CNT
118- A5 EA 0440 LDA *OUTS :GET COPY PINKING
11A- 06 EA 0450 DEC *OUTS :COUNTER, DEC ORIGINAL
11C- 45 EA 0470 EOR *OUTS :PATTERN OF CHANGED
11E- 85 EB 0490 STA *OUTT :BITS - SAVE CHANGES
120- A9 00 0500 LDA 00 :PREPARE TO SUM DICE
122- 46 EB 0510 NM1 LSR *OUTT :CHECK FOR CHANGED
124- 90 0A 0520 BCC NM2 :BIT - IF CHANGED,
126- 48 0530 PHA :SAVE CURRENT TOTAL
127- 20 00 01 0540 JSR RND :GET RANDOM NUMBER

```

```

12A- 29 03 0550 AND 03 :MAKE RANGE 0 TO 3
12C- 95 9F 0560 STA *RAND,X :SAVE VALUE FOR NEXT
12E- 00 0570 PLA :RECOVER TOTAL
12F- 18 0580 CLC :PREPARE ADDITION
130- 75 3F 0590 NM2 ADC *RAND,X :ADD VALUE OF DICE
132- CA 0600 DEX :POINT TO NEXT
133- 00 ED 0610 BNE NM1 :LOOP IF NOT DONE
135- AA 0620 TAX :USE TOTAL AS POINTER
136- 85 90 0630 LDA *NBUFF,X :GET CANDIDATE
138- F0 03 0640 BEQ DUR0 :ZERO, DO NOT CHANGE
13A- 99 BF 00 0650 STA NTB7,Y :PLACE IN TEMP BUFFER
13D- A5 A5 0660 DUR0 LDA *NOIS+01 :A CHEAP RANDOM NO.
13F- 18 0670 CLC :PREPARE
140- 39 88 00 0680 AND MASK,Y :MASK DURATION VAL.
143- 79 87 00 0690 ADC TIME,Y :ADD MINIMUM VAL.
146- 29 0F 0700 AND 0F :AND MASK RESULT
148- AA 0710 TAX :USE AS COUNTER AND
149- A9 01 0720 LDA 01 :DO DURATIONS AS
14B- 2A 0730 NT2 ROL :POWERS OF 2, CARRY
14C- CA 0740 DEX :SET DOTS NOTE
14D- 00 FC 0750 BNE NT2 :NOT DONE - LOOP
14F- 99 C3 00 0760 STA NTB8,Y :PUT RESULT IN NOTES
152- 60 0770 RTS :TIMER AND RETURN
      0780 :
      0790 : ALLOCATION 0151
      0791 :
      0792 : SEES IF NEW NOTES ARE NEEDED AND IF
      0793 : SO GETS THEM. ALSO CLEARS TRIGGER
      0794 : OF NOTE OUTPUT ONCE IT IS PLAYED.
      0800 :
153- A2 04 0810 ALOC LDX 04 :DO 4 NOTE CHANNELS
155- D6 C3 0820 LP6 DEC *NTB8,X :DECREMENT NOTE TIMER
157- D0 07 0830 BNE LP5 :AND IF TIME OUT
159- 0A 0840 TXA :TRANSFER X REG. TO
15A- A8 0850 TAY :TO Y
15B- 20 16 01 0860 JSR NEW :AND GET NEW NOTE
15E- 98 0870 TYA :AND DURATION AND
15F- AA 0880 TAX :RESTORE X
160- CA 0890 LP5 DEX :DECREMENT COUNTER
161- D0 F2 0900 BNE LP6 :IF NOT DONE - LOOP
163- A2 04 0920 LDX 04 :AGAIN, FOUR CHANNELS
165- B5 BF 0930 AL1 LDA *NTB7,X :GET NOTE FROM TEMP
167- 95 D8 0940 STA *NT08,X :BUFFER, SAVE IN OUT
169- 29 3F 0950 AND 3F :BUFFER, CLEAR FLAG
16B- 95 BF 0960 STA *NTB7,X :PUT BACK IN TEMP.
16D- CA 0970 DEX :POINT TO NEXT
16E- D0 F5 0980 BNE AL1 :NOT DONE - LOOP
170- 60 0990 RTS :DONE, RETURN
      1000 :
      1010 : SET
      1011 :
      1012 : PREPARES KNOWN STARTING POINT FOR
      1013 : CYCLIC TUNES.
      1014 :
      1020 :
171- A9 00 1030 SET LDA 00 :TO ZERO THINGS WITH
173- A0 01 1040 LDY 01 :PRESET FOR NOTE CNTRS
175- A2 04 1050 LDX 04 :DO 4 CHANNELS
177- 95 D8 1060 LP10 STA *NT08,X :ZERO OUT-BUFFERS
179- 95 A0 1070 STA *RND0,X :ZERO 4 DICE
17B- 94 C3 1080 STY *NTB8,X :PRESET NOTE TIMERS
17D- 48 1090 PHA :SAVE THE ZERO
17E- B5 CF 1100 LDA *NTMP,X :SET UP RNDM'S S/R
180- 95 A4 1110 STA *NOIS,X :AND CYCLE COUNTER
182- 68 1120 PLA :RECOVER ZERO
183- CA 1130 DEX :POINT TO NEXT
184- D0 F1 1140 BNE LP10 :NOT DONE - LOOP
186- 85 A0 1150 STA *RND0 :ZERO 5TH DICE
188- 85 EA 1160 STA *OUTS :ZERO PINKING COUNTER
18A- 60 1170 RTS :AND RETURN
      1180 :
      1190 END EN

```