

OSound2 manual

version 2.0

- 0. New in this version
- 1. Introduction
- 2. Installation
- 3. Quick guide for OSound2
 - 3.1 Manual mode - no cycle limit
 - 3.2 Manual mode - with cycle limit
 - 3.3 Auto mode
- 4. Menus
 - 4.1 Matrix parameters
 - 4.2 Set up matrices
 - 4.3 Cycle parameters
 - 4.4 Manual mode
 - 4.5 Auto mode
 - 4.6 Dynamic
- 5. Operation
- 6. Further processing of data
 - 6.1 The file naming system
 - 6.2 The ntof program
 - 6.3 The midigen program
 - 6.4 The csvmidi program
- 7. Some important comments
- 8. Further development
- 9. Bug reports

0. New in this version

An entire text can be read into the system from a file. Each character represents an input.

Sections that have been modified:

- 4.5 Auto mode
- 5. Operation

1. Introduction

OSound2 is a derivative of OtoomCM (the acronym comes from the associated book's title "On the origin of Mind" and Computer Model) which represents a prototype artificial mind. Its fundamental dynamic consists of the creation, maintenance or destruction of clusters of fitness peaks through forming affinity relationships with other clusters which in subsequent cycles become more or less confirmed.

OSound2 allows the user to generate a text file containing all the integers necessary to create a midi file from the matrix values using the other three programs *ntof*, *midigen* and *csvmidi* (see **7. Further development**).

To learn about the details behind OtoomCM's dynamic consult the Appendix in the Otoom book first, then chapters 14 and then 15 for a still broader context. This manual is concerned with the user interface of OSound2 only.

2. Installation

OSound2 is a DOS-based program, written in C/C++. Click on **Oso2.exe** and the program starts in a full-screen DOS window under Win98, Win2000 and XP. (If it doesn't, right-click on the program > Properties > Screen > Usage and click Full-screen)

No changes are made to the hard drive, the operating system, or anywhere else.

3. Quick guide for OSound2

3.1 Manual mode - no cycle limit

- Click on **Oso2.exe** to start the program.
- Press **x** to check the maximum parameter values.
- Press **m** to set your choice of matrix parameter values.
- Press **s** to set up the matrices.
- Press **c** to set your choice of cycle parameter values.
- Press **n** to set the program to manual, making sure *cycle numbers* is left at '0' and the output text file is given a name (no file extension, maximum 8 characters).
- Press **d** to start the matrix operations.
- Press any key(s) during the cycles to provide input.
- Press **Esc** at any time to stop the operation.

3.2 Manual mode - with cycle limit

- Click on **Oso2.exe** to start the program.
- Press **x** to check the maximum parameter values.
- Press **m** to set your choice of matrix parameter values.
- Press **s** to set up the matrices.
- Press **c** to set your choice of cycle parameter values.
- Press **n** to set the program to manual, set *cycle numbers* to the desired value (at least '1') and give the output text file a name (no file extension, maximum 8 characters).
- Press **d** to start the matrix operations.
- Press any key(s) during the cycles to provide input.
- The program will stop when the given cycle number has been reached.

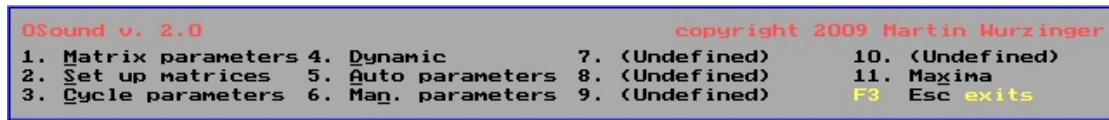
3.3 Auto mode

- Prepare the input parameter text file, making sure the name agrees with the name appearing in its header (maximum 8 characters, file extension = .txt).
- Click on **Oso2.exe** to start the program.
- Press **x** to check the maximum parameter values.
- Press **m** to set your choice of matrix parameter values.
- Press **s** to set up the matrices.
- Press **c** to set your choice of cycle parameter values.

- Press **a** to set the program to auto, and type in the input text file name of the prepared file (no file extension, maximum 8 characters).
- Press **d** to start the matrix operations.
- The program will stop when the given cycle number has been reached.

4. Menus

The menu section:



Select a menu by pressing the underlined letter on the keyboard.

F3 and Esc exits the program and/or a menu and either one is operational when highlighted.

If no matrix has been defined, first **Matrix parameters** must be selected, then **Set up matrices**, then **Cycle parameters**, otherwise error messages will result.

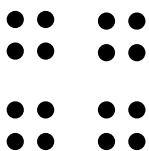
4.1 Matrix parameters

Press **m** on the keyboard.

Input window for **Matrix parameters**:

node matrix rows:	<input type="text" value="24"/>	cols:	<input type="text" value="24"/>
elem matrix rows:	<input type="text" value="10"/>	cols:	<input type="text" value="10"/>
connect'y/node %:	<input type="text" value="5"/>	KB :	<input type="text" value="323"/>
high end of elem. number range:	<input type="text" value="200"/>		

The system's dynamic operates within a matrix where each individual entry is another matrix. The matrix entries are referred to as nodes, and the entries of a node are called elements. For example:



The above represents a matrix with four nodes and four elements per node. Or, a matrix with two node rows and two node columns, and two element rows and two element columns.

WARNING: Due to DOS memory limitations under the Windows 98, 2000 and XP operating systems the maximum safe values are those shown under **11. Maxima**. Anything higher may crash the O/S (there are no technical limits to the values in terms of the program itself).

Enter the values for node matrix rows etc by pressing Tab to proceed left-right, top-down; for the opposite direction use Shift-Tab. Press Enter to exit each field and store the value.

Hit **Esc** to save the parameters and exit the menu.

Notes:

connect'y/node %: the number of nodes expressed as a percentage of the total number of nodes which every single node is connected to. Example: a value of 10% with 100 nodes in the matrix means that each node is connected to 10 other nodes.

KB: the memory used in kilobytes and calculated automatically when the respective values have been entered. The value is affected by every parameter.

high end of elem. number range: the highest integer which will be used for the node elements (the range starts with 0). The higher this number, the more expressive the output will be. The number does not affect the memory usage.

4.2 Set up matrices

Press **s** on the keyboard to set up the matrix according to the parameters entered before.

Hit **Esc** to save the parameters and exit the menu.

The memory indicator on the bottom of the screen shows the memory used (red bar) and the amount of memory left in KB as a number (green box):



Ongoing date and time values are displayed on the left and right side.

4.3 Cycle parameters

Entering these numbers follows the same Tab system as for the matrix parameters.

input regions:	<input type="text" value="2"/>	output regions:	<input type="text" value="2"/>
		n. disp. freq.:	<input type="text" value="10"/>
block factor :	<input type="text" value="1"/>	conn. depth :	<input type="text" value="2"/>

Hit **Esc** to save the parameters and exit the menu.

Notes:

input regions: becomes the total number of input nodes, divided by the entered figure.

output regions: becomes the total number of output nodes, divided by the entered figure.

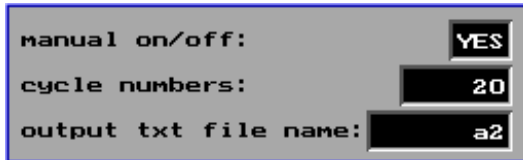
n. disp. freq.: update frequency of the display on the screen showing the state of affinity relationships for each node as the system goes through its cycles. Example: a value of ten means that the display will be updated every 10th cycle. HINT: calculating this display uses by far the most resources. If not required at all, set to 0.

block factor: the number of times a depth traversal is repeated.

conn. depth: the number of depth levels to be traversed.

4.4 Manual mode

Press **n** on the keyboard to select manual mode, the desired number of cycles before the operation stops (if any), and the output text file a name (no file extension, maximum 8 characters).



A screenshot of a terminal window showing the 'Manual mode' menu. It has three rows: 'manual on/off:' with a 'YES' button, 'cycle numbers:' with a '20' input field, and 'output txt file name:' with an 'a2' input field. The entire menu is enclosed in a blue border.

Tab into *manual on/off* to toggle from NO to YES and vice versa.


Tab into the next field to select the desired number of cycles the program is meant to run. If no set cycle limit is desired set *cycle numbers* to '0'.

Tab into the next field and choose a name for the output text file.

Hit **Esc** to save the parameters and exit the menu.

4.5 Auto mode

Used for automated runs, where various cycle parameters are input from a file.



A screenshot of a terminal window showing the 'Auto mode' menu. It has two rows: 'auto on/off:' with a 'YES' button, and 'input txt file name:' with a 't1' input field. The entire menu is enclosed in a blue border.

Tab into *auto on/off* to toggle from NO to YES and vice versa.

Tab into the next field to type the name of the input parameter text file (maximum 8 characters, no file extension).

Example of a parameter input file, with user-defined values in bold:

```
otoomt
input_frequency:
1
display_region:
1
block_factor:
1
connection_depth:
5
node_display_frequency:
1
initial_seeding:
```

```

0
seeding_value:
0
seeding_frequency:
0
element_writing:
1
text:
Hello world!

```

Notes:

input_frequency: how often in cycles a basic input will be presented to the system.

display_region: denotes the in- and output region that should be active during the cycles.

block_factor: the number of times a depth traversal is repeated.

connection_depth: the number of depth levels to be traversed.

node_display_frequency: update frequency of the display on the screen showing the state of affinity relationships for each node as the system goes through its cycles. This value must be set to '1' to generate the affinity values at every cycle used later.

initial_seeding: '0' off, '1' on. When 'on' an input value is presented to the system before the actual evaluation begins.

seeding_value: input integer used for seeding.

seeding_frequency: number of times the seeding value is presented to the system.

element_writing: for writing the element values to a text file. Must be set to '1'.

text: Add any text you wish here.

NOTE: depending on the size of the matrix and/or the number of elements per node as well as the number of cycles required and the text used, this file can get quite large.

WARNING: Do not alter the format of the parameter input file. Use '0' to cancel a parameter.

4.4 Dynamic

Press **D** to prepare the system to start. To actually commence the cycles press the Spacebar.

Using this command to start and/or recommence the cycles will completely reset all the screen displays. To continue after a Pause (Ctrl-Z) with the current matrix content and screen values preserved, use the Spacebar only.

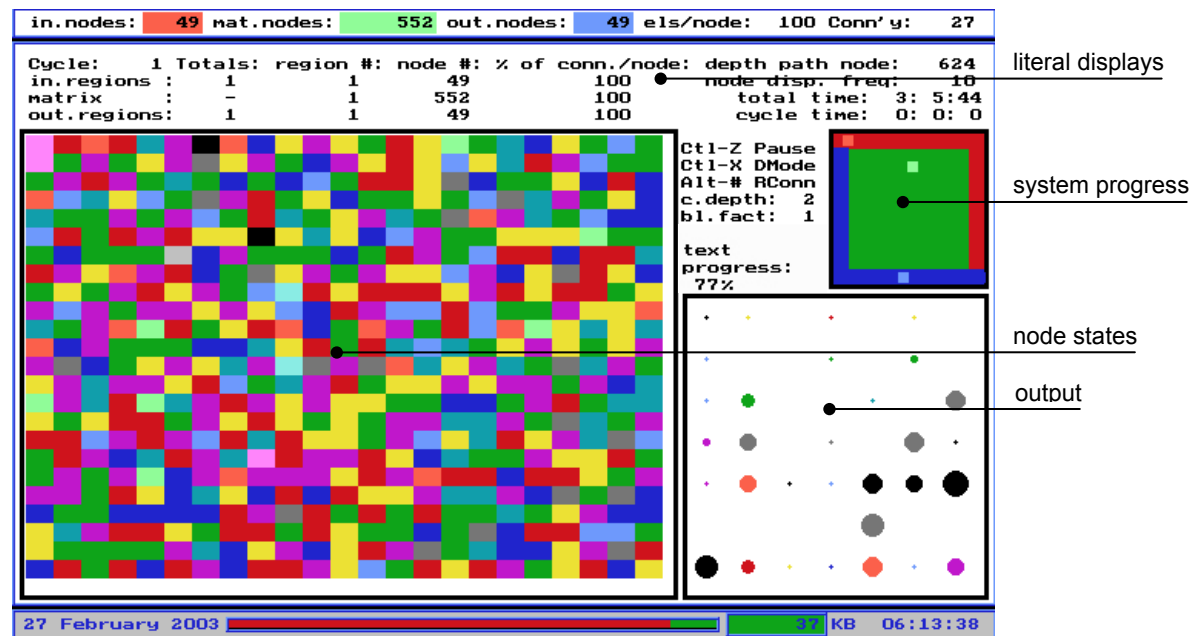
5. Operation

Operation is started by pressing the Spacebar. The system will cycle continuously, regardless of the presence of input or output. During cycling, pressing any key on the keyboard which is not designated for a particular purpose stops the current cycle and starts it once again with the

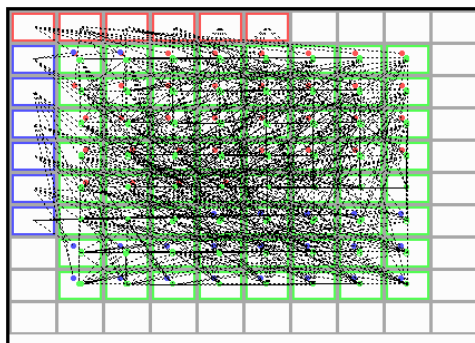
new input value (ie, the integer value of that key). To stop the cycles completely, press Esc to exit the menu.

To pause the cycles press Ctrl-Z, to restart press the Spacebar.

When displaying the affinity relationships, the screen looks something like this:



If the update frequency of the display has been set to 0 when setting the cycle parameters, no node states will be shown. If set to 1, Ctr-X will toggle between a display of affinity relationships for every node (shown above under *node states*) and the layout plan of the matrix, showing the active input region (red), the active output region (blue), the processing nodes of the matrix (green), and the connections between all the nodes. For example:



Notes:

in.nodes: number of input nodes

mat.nodes: number of inner matrix nodes

out.nodes: number of output nodes

els/node: number of elements per node

Conn'y: number of nodes connected to each node

Cycle: number of current cycle

Totals: number of total in- and output regions

region #: current region being processed (incl matrix)

node #: current node being processed

% of conn./node: percentage of node+tree processed

depth path node: node in the tree traversed

node disp. freq: display frequency in terms of cycles

total time: time taken for one test run

cycle time: time taken for one cycle

Ctl-Z, Ctl-X: real-time toggle switches for Pause, Display Mode (node state or connections)

Alt-#: real-time selector for in- and output region by number (#)

c.depth: number of depth levels to be traversed

bl.fact: number of times a depth traversal is repeated (block factor)

text progress: how much text read in from the input file, expressed as percent of the total

Bottom section: date, memory used of available total (bar), memory left (integer), time

6. Further processing of data

After having run OSound2 a text file has been generated that allows further processing of the data. The associated programs are *ntof*, *midigen* and *csvmidi*.

6.1 The file naming system

It is assumed the name of the file is *a.txt* (which is either the input parameter file for auto mode or the name of the output file under manual mode - both typed in as 'a', ie no file extension).

a.txt - used by OSound2, produces..
a-e.txt - used by ntof, produces..
a-f.txt - used by midigen, produces..
a-m.txt - used by csvmidi, produces..
a.mid

6.2 The ntof program

Assuming the name of the input file is *a-e.txt*, the command line is
`ntof a-e`

The `ntof` program (“number to frequency”) maps the integers produced by each node of the inner matrix to a frequency that applies to an actual note (every number can be seen as a frequency, but not every frequency applies to an actual musical note).

To this end the integers produced by each node of an element matrix (every node of the inner matrix represents another matrix, the element matrix) are averaged so that each inner matrix node is represented by one integer. If some of the integers happen to be negative, the entire set is shifted up by the maximum negative value (or the minimum overall integer value) achieved for that particular run under `OSound2` while at the same time reflecting the next integer that is mappable to an actual note.

The result is `*-f.txt`, which contains a summary of the data and the integers for each node plus the average affinity value for each cycle.

Example of an input file (`*-e.txt`):

```
cycle 1
node 1
10000 9842 9684 9526 9368 9210 9052 8894 8736 8578 8420 8262 8104 7946 7788
7630 7472 7314 7156 6998 6840 6682 6524 6366 6208 6050 5892 5734 5576 5418
5260 5102 4944 4786 4628 4470 4312 4154 3996 3838 3680 3522 3364 3206 3048
2890 2732 2574 2416 2258 2100 1942 1784 1626 1468 1310 1152 994 836 678 520
362 204 46
Affinities 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
node 2
10000 9842 9684 9526 9368 9210 9052 8894 8736 8578 8420 8262 8104 7946 7788
7630 7472 7314 7156 6998 6840 6682 6524 6366 6208 6050 5892 5734 5576 5418
5260 5102 4944 4786 4628 4470 4312 4154 3996 3838 3680 3522 3364 3206 3048
2890 2732 2574 2416 2258 2100 1942 1784 1626 1468 1310 1152 994 836 678 520
362 204 46
Affinities 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
node 3
10000 9842 9684 9526 9368 9210 9052 8894 8736 8578 8420 8262 8104 7946 7788
7630 7472 7314 7156 6998 6840 6682 6524 6366 6208 6050 5892 5734 5576 5418
5260 5102 4944 4786 4628 4470 4312 4154 3996 3838 3680 3522 3364 3206 3048
2890 2732 2574 2416 2258 2100 1942 1784 1626 1468 1310 1152 994 836 678 520
362 204 46
Affinities 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
node 4
10000 9842 9684          ... etc.
```

Example of an output file (`*-f.txt`):

```
File                : awl0-f.txt
number of cycles    : 10
number of nodes     : 144
min matrix element value : -14376
max matrix element value : 21748
min frequency value  : 40
max frequency value  : 14000
values adjusted up by : 14416
max affinity value   : 17
min affinity value   : -74
NOTE: '|' denotes the end of one row per cycle.
```

```
536 536 536 536 536 536 536 536 536 536 536
536 536 536 536 536 536 536 536 536 536 536
536 536 536 536 536 536 536 536 536 536 536
536 536 536 536 536 536 536 536 536 536 536
536 536 536 536 536 536 536 536 536 536 536
536 536 536 536 536 536 536 536 536 536 536
```

536	536	536	536	536	536	536	536	536	536	536
536	536	536	536	536	536	536	536	536	536	536
536	536	536	536	536	536	536	536	536	536	536
536	536	536	536	536	536	536	536	536	536	536
536	536	536	536	536	536	536	536	536	536	536
536	536	536	536	536	536	536	536	536	536	536
536		3								
536	536	536	536	536	536	536	536	536	536	536
536	536	536	536	536	536	536	536	536	536	536
536	536	536	536	536	536	536	536	536	536	536
536	536	536	536	536	536	536	536	536	...	etc.

6.3 The midigen program

Assuming the name of the input file is *a-f.txt* and the ensemble number chosen is '1' (see below), the command line is

```
midigen a-f 1
```

and so its output file will be *a-m1.txt*.

The midigen program ("midi generator") takes the output file generated by ntof and turns it into a text file that can be used by the csvmidi program.

For an explanation of all the midi parameters see <http://www.fourmilab.ch/webtools/midicsv/>.

Each cycle is now taken as a separate track. To abide by the midi format the times at which each note needs to be played as well as stopped are attached to every on-note and/or off-note. If an integer is repeated after each other it is mapped to one single note, its length reflecting the number of repetitions.

In addition to the note value already determined, the value is also changed according to the length arrived at as well as the affinity values generated within the inner matrix of OSound2.

Notes need to be attached to instruments. The following guide lines were used:

- Instruments that feature a long sustain value (as part of the entire set of attack, sustain, decay) under the midi interpretation of that instrument are given to long notes (eg, organ).
- Instruments that feature a not so long sustain value are given to shorter notes (eg, piano).
- Instruments that feature a short sustain value are given to short notes, ie those that come from a single occurrence of an integer (eg, woodblock).

Various instruments have been grouped under ensembles (the following list contains the instruments and their numbers under the midi format). Presently there are seven, namely

1. 43=cello, 72=clarinet, 74=flute, 69=oboe, 1=acoustic grand piano, 2=bright acoustic grand piano
2. 43=cello, 72=clarinet, 74=flute, 70=english horn, 42=viola, 41=violin
3. 43=cello, 42=viola, 41=violin, 1=acoustic grand piano, 2=bright acoustic grand piano
4. 75=recorder, 28=electric clean guitar, 29=electric guitar muted, 36=fretless bass, 13=marimba
5. 54=ooh choir, 31=distortion guitar, 121=guitar fret noise, 46=pizzicato strings, 108=koto, 10=glockenspiel, 116=woodblock
6. 17=draw organ, 18=percussive organ, 11=music box, 102=goblin, 97=ice rain, 77=bottle blow, 10=glockenspiel

7. 86=synth lead 6, 89=synth pad 1, 78=shakuhachi, 91=synth pad 3, 92=synth pad 4, 119=synth tom, 14=xylophone

Hence the command line for midigen includes the number of the chosen ensemble group.

Each instrument is given its particular channel.

Example of an output file (**-m*.txt*):

```
# filename:  awl0-m1.txt
# This midi file has been created from
# the output of OSound2, an artificial mind
# prototype program.
# The integer values from its internal
# processes have been mapped to musical
# note values which are transposed to a
# midi csv file format.
0, 0, Header, 1, 11, 384
1, 0, Start_track
1, 0, Copyright_t, "Copyright (C) Martin Wurzinger"
1, 0, Time_signature, 1, 2, 24, 8
1, 0, Key_signature, 0, "major"
1, 0, Tempo, 350877
1, 1, End_track
2, 0, Start_track
2, 0, Pitch_bend_c, 0, 40
2, 0, Control_c, 0, 7, 100
2, 0, Pitch_bend_c, 1, 8192
2, 0, Control_c, 1, 7, 100
2, 0, Pitch_bend_c, 2, 8192
2, 0, Control_c, 2, 7, 100
2, 0, Pitch_bend_c, 3, 8192
2, 0, Control_c, 3, 7, 100
2, 0, Pitch_bend_c, 4, 8192
2, 0, Control_c, 4, 7, 100
2, 0, Pitch_bend_c, 5, 8192
2, 0, Control_c, 5, 7, 100
2, 0, Program_c, 0, 43
2, 0, Title_t, Cello
2, 0, Program_c, 1, 72
2, 0, Title_t, Clarinet
2, 0, Program_c, 2, 74
2, 0, Title_t, Flute
2, 0, Program_c, 3, 69
2, 0, Title_t, Oboe
2, 0, Program_c, 4, 1
2, 0, Title_t, Acoustic grand piano
2, 0, Program_c, 5, 2
2, 0, Title_t, Bright acoustic piano
2, 384, Note_on_c, 0, 0, 41
2, 1536, Note_off_c, 0, 0, 41
2, 56064, Note_on_c, 1, 69, 42
2, 222720, End_track
3, 0, Start_track
3, 0, Pitch_bend_c, 0, 40
3, 0, Control_c, 0, 7, 100
3, 0, Pitch_bend_c, 1, 8192
3, 0, Control_c, 1, 7, 100
3, 0, Pitch_bend_c, 2, 8192    ... etc
```

6.4 The csvmidi program

Assuming the name of the input file is *a-m1.txt* and the desired name for the output file is *a-1.mid* the command line is
`csvmidi a-m1.txt a-1.mid`

For an explanation of all the midi parameters see <http://www.fourmilab.ch/webtools/midicsv/>. (Also available from this website is its opposite, *midicsv*, which turns a midi file into a text file similar in format to the example under 6.3)

The resultant midi file is now a binary file, ready to be played by a synthesiser.

NOTE: ‘csv’ stands for ‘comma separated values’.

For further notes see <http://www.otoom.net/SoundOfMind.htm>.

7. Some important comments

This is no ordinary computer program.

Due to its very nature it is highly sensitive to input, meaning the input buffer is polled constantly and any value there is considered a valid input. Especially during auto mode do not press any keys as their values will most certainly find their way into the matrix, to be processed there. This even applies to Alt-Tab to change the focus on windows!

Because the output is a result of the numerical relationships between nodes, even the slightest change in a value in some element matrix influences the outcome (the old butterfly effect again). Since ultimately the presence of any such value relies on the hardware (ie, RAM, bus, motherboard, CPU registers, and all the way back to memory and the program) even a minute change in the steps by clock ticks will alter the effective presentation of a certain value to its respective matrix.

This means an output derived from a run in a certain directory can be different from the output after having copied the program to another directory!

This is also the reason why I didn’t include the means to log the key presses under manual mode. Yes, having hit upon a really interesting output and therefore soundscape would want you have some kind of log to remember the key presses for posterity, but alas, that’s not possible. Even if the inputs were logged (and this is possible), in order to recreate the *exact* timing of those presses their times would need to be logged as well. Although in essence this is also possible, to feed those back during another run the ongoing timeline needs to be polled and the relevant clock ticks responded to. This means further processing, so the actual clock ticks are now changed which means the input arrives at different times, and in the end the result will be different anyway. Sorry!

Obviously the above does not apply to *ntof*, *midigen*, or *csvmidi*.

8. Further development

This is just an exercise intended to demonstrate that the numbers generated within the matrix of the AI program can be interpreted as frequencies and/or notes. It is certainly not meant to lay any claim to what people may understand by ‘music’!

On the other hand... suppose the AI program’s capacity was extended to receive any type of input, such as visuals via a camera. Such an input could be turned into a soundscape (with

possible sweetening applied) provided a frequency generator is attached which receives the program's output and returns the sound in real-time.

The richness of such a soundscape is only limited by the size of the AI system, ie the size of its matrices. The program would have to be transposed into a distributive format to enable real-time processing.

9. Bug reports

To my knowledge the program is bug-free - but you never know!

If you encounter any problem please contact me via <http://www.otoom.net/Contact.htm>.

To be of any help I need to know all the parameter settings under the menus **Matrix parameters**, **Cycle parameters**, **Auto parameters** and **Man. parameters**, plus the input parameter text file if applicable.