AlgoMusic

COLLABORATORS			
	TITLE : AlgoMusic		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AlgoMusic

1.1   Welcome to the AlgoMusic documentation

                              AlgoMusic V1.4

                   (C) 16.10.95 by Thomas Schürger

                   A nice algorithmic song player

                             Giftware

                             Contents

                    Copyright

                  Introduction

                    Changes

                  Requirements

                     Usage

                  Bugs/Problems

                     Future

                    Credits

                    Author
                          Note: My main e-mail-address has not changed yet!

              Please rate AlgoMusic on Aminet! See "Author"!

1.2   Copyright

  AlgoMusic is Giftware. This means that you can copy, use or even delete it

in any manner you like. But you mustn't modify ANY of the files supplied in
the AlgoMusic-archive, crunching with any packer available to the public is
allowed, though. You may not gain any commercial or private profit out of
this program.

   You are asked to send me a gift (disks, modems, Ferraris, swimming pools
and F-18s are preferred..) if you like AlgoMusic.

   AlgoMusic may be taken for use in any PD/Freeware/Shareware library
(including Aminet and Fred Fish's library), but I would like to be informed
about it via e-mail. It would be great if you could send me a copy of the
disk/CD it is published on.

   You are asked to upload AlgoMusic on every BBS you see! B)

1.3 Introduction

   Ever been tired of looking into your soundmodule-directory and yawning
your head off because you've heard all your mods thousands of times? Well,
here is the ultimate solution!

   AlgoMusic is a little (?) program that creates and plays a song
randomly out of a series of algorithms. Just try it, it sounds nice. :)

   It first creates the song (which is very fast), then switches to a
task priority of 25 (it needs as little CPU-time as possible) and switches
off the filter (if not disabled), then plays and restores everything
afterwards. It is safe to break AlgoMusic at any time. Another way to
end AlgoMusic is to start AlgoMusic again from another task (arguments
except QUIET then are ignored).

1.4 Requirements

   AlgoMusic should (!) run on any Amiga with at least 512 KB of chipmem and
512 KB of fastmem. At least Kickstart 2.0 (V36) is required.

   It was developed on a PAL Amiga 1200 with 68030/50. It has also been
tested on an A500+, but never thoroughly on a 040/060 processor. If you have
any problems (especially with CPU-caches left on on those processors)
getting it run on your own machine, please let me know.

1.5 Usage

   Upto this stage, AlgoMusic can only be started via Shell or using
"Execute Command" via Workbench.

   The argument-template looks as follows:

     SPEED=DELAY/K/N,FOREVER/S,QUIET/S,FILTER/S,NOFADE/S,VOLUME/K/N,ACID/S,

```
   INIT=RANDOMINIT/K,BPM/K/N,NOCIA/S,NOALLOC/S,BREAKBEAT/S
```

SPEED=DELAY n   This is the speed the created song is played at. The
                     lower the number, the faster it is played. The
                     standard speed is 4. The speed must be in the range
                     of [2..8]. It will be ignored if you don't use NOCIA.

FOREVER=REPEAT   If you want AlgoMusic to go on playing forever, this
                     is the right switch to use. After finishing a song,
                     a new song is quickly created and played. By default,
                     FOREVER is switched off.

QUIET   All output is suppressed, except error messages.

FILTER   If this switch is activated, then the audio filter is
                     switched on. By default, the filter is switched off.

NOFADE   Normally, the song is smoothly faded out in the
                     song's last pattern. Using this switch, you can
                     disable this.

VOLUME n   This sets the volume the song is played at. It must
                     in the range of [0..64]. Default is 64 (maximum
                     volume).

ACID   Switches on the use of an acid-style sample where
                     applicable. By default, it is switched off.

INIT=RANDOMINIT $h   Initializes the random generator with the hex-value
                     specified. Same hex numbers will result in same
                     songs.

BPM n   Sets the playing speed to the BPM value specified.
                     It will be ignored if you use NOCIA, because then,
                     DELAY/SPEED will be used. It must be in the range of
                     [84..250]. The default is 150.

NOCIA   If you don't want AlgoMusic to use a CIA timer, you
                     can tell it to use Delay() instead. Then the
                     DELAY/SPEED value will be used for setting the playing
                     speed.

NOALLOC   This switch disables the allocation of audiochannels.
                     By default, the channels are allocated.

BB=BREAKBEAT   Enables using breakbeats where applicable.

Examples:

1> AlgoMusic BPM 140

would play it a bit slower than with the default value.

If you like it fast and forever with acid-style and breakbeats, you
```

might use

  1> AlgoMusic BPM 168 ACID FOREVER BREAKBEAT                    :^)


   You can put your favourite arguments into a variable called
"ENV:AlgoMusic.prefs". It is parsed just the way the shell arguments
are parsed. Shell arguments override the variable arguments, which means
that if you have specified "INIT $4002" in the variable and use "INIT $5"
in the shell arguments, the second value is taken.
   Switches are handled in another way. If you specify the same switch in
both the variable and the shell arguments, it is switched off, because it
is first turned on and toggled back off afterwards.

   If you already have an AlgoMusic process running, you can stop it either
by sending a break signal to the process or by just running AlgoMusic again.


## 1.6  Changes

   Dates are dates of release as well as dates of last change.

Changes since V1.0 (12-Apr-95 at 1 a.m.):


V1.1:  (12-Apr-95 afternoon)   Minor Update
-------------------------------------------
  o Added the ability to choose playing speed per argument.

  o Used better rhythm samples (basedrum, clap).

  o Added other chords, basses~& melodies.

  o Added version-string :)

  o Added more freedom for creating the song-structure.


V1.2:  (18-Apr-95)   Major Update
---------------------------------
  o Added optional fadeout at the end of the song.

  o Now Kick 2.0 (V36) is required (!).

  o Added proper argument handling by using ReadArgs().

  o Added new switches: FOREVER, QUIET, FILTER and NOFADE.

  o Added the ability to set VOLUME.

  o Added some more rhythms, chords, basses and melodies.

  o Enhanced random generator.


V1.3:  (13-May-95)   Major Update

```
--------------------------------
   o Included a nice CIA timer for playing.

   o Added option NOCIA which plays just the former way (using Delay()).

   o Added the ability to set the BPM rate per argument.

   o Added acid-style sample and option ACID.

   o Now outputs the random seed.

   o Added the ability to set the random seed as a hex value via
     RANDOMINIT/INIT argument.

   o Found out how to play a sample without automatic repetition... ;)
     The samples all used to have quite a big zero-byte buffer behind them.
     This has been changed -> AlgoMusic is MUCH shorter now!

   o Rewrote the song creation algorithm. There used to be only four chords
     per pattern and fixed bass/melody sequences (16 steps long). This has
     been changed. Now the chords can span over several patterns and can be
     changed quicker etc.

   o Added the ability to put one's favourite arguments into a variable
     called "ENVARC:AlgoMusic.prefs". Look at the "Usage" section to find
     out more about this.

   o Now, "INIT $0" is possible. It turns the pseudo-random random seeding
     back on. Not bad if you have specified an init-value in the ENV-variable
     and want to specify in the shell arguments to have real randomization
     again.

   o Rhythms don't sound as monotonous as before.

   o Now allocates audiochannels. May be disabled using NOALLOC.

   o Now songs have names. >;-)

   o Switched to an AmigaGuide documentation.


V1.4:  (16-Oct-95)   Major Update
--------------------------------
   o Added breakbeats! I don't know anymore who proposed that, but I liked
     the idea! Can be enabled using the BREAKBEAT switch.

   o Small bug: BPM rate could not be set below 84.

   o Changed to Giftware!
```

1.7 Bugs/Problems

  There are still some bugs and problems left:

o The player still does not run 100% smooth when certain programs are
  used. Those seem to use Disable()/Enable() while scrolling. Examples are
  FinalWriter and Mand2000Demo. This does rarely happen and I can't change
  it. If you find more, tell me!

o Uses direct hardware access on audiochannels.

o Especially the acid-style sample doesn't sound very nice sometimes. If
  you have a better one, let me know!

o You will find that the RANDOMINIT values are useless when using
  different versions of AlgoMusic. You'll get other songnames as well.

o In the beginning when using breakbeats, those sometimes used to sound
  strange. If this happens somewhere again, tell me the init-number of
  that song!

They will be (partly) removed in the future. So watch out for further
updates!

1.8 Future

I will continue working on AlgoMusic in the future, because I have already
received a lot of feedback. Of course, some gifts would morally support me
as well! ;-)

Many people have suggested to make AlgoMusic save a created module in
protracker-format. This is not what AlgoMusic is supposed to do. My replayer
is much shorter than any protracker-replayer, and AlgoMusic itself is shorter
than the resulting module would be, so I will NOT include that in the future!

I will, of course, add more customization like randomly choosing the used
samples out of a directory and so on. Be prepared for it! :)

1.9 Credits

I would specially like to thank the following people for their support
and help:

- Rüdiger Prang   o My betatester.

- Kari-Pekka Koljonen   o Supplied me with a nice CIA-timing-routine.

- Richard Körber   o Reported some bugs, sent me a source for
                           audiochannel allocation, publishes AlgoMusic
                           on Aminet.

- Mike Witte   o Checked '040 compatibility.

- Felix Müller   o Reported a bug.

Thanks to all the other people that sent me a mail about previous
versions!

1.10 Author

If you like AlgoMusic, or even if you don't (but then let me know why),
if you have any suggestions, questions or ideas, or if you have found
any bugs (even in this guide), send a mail to:

     Phobos@eratosth.dssd.sub.org
  or: T.Schuerger@proline.gun.de (preferably)

My e-mail-domain will change (only the Proline-address is
affected). My new address from then on will be:

     T.Schuerger@proline.ndh.com

I don't quite know when this will happen (if it does anyway). So please
try contacting me using my old address first!

Please rate AlgoMusic on Aminet!

Simply send a message with no subject to: Aminet-Server@aminet.org
Contents: rate mus/misc/AlgoMusic1_4.lha <number>
where <number> is a number from 1 to 10, 10 is the best.

For those still not being able to send E-mails or those who would like to
send me gifts:

My snail-mail-address:

Thomas Schürger
Birkenbusch 39
53757 Sankt Augustin
Germany

Contact me, I would be very pleased to receive a lot of feedback.

My PGP-key: (use it if possible!)

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.2i

mQCNAi6SRPYAAAEEAMHGFQM/HLl5g+IoORD+ErAru7Dd17S1OkqyD2t6qWDGjTPE
ym7bd1ESLoWeXtfNiqvkbdeV9gELmFZ8cT5uKicesCMZkOtIuu23rDLccIXClQU+
RU1sVsTAub2Gm4RSg15XrMipqC5Waf/+QAlwlq6u5qtjVgMAMys4JdI2NDr5AAUR
tC1UaG9tYXMgU2NodWVyZ2VyIDxULlNjaaHVlcmdlckBwcm9saW5lLmd1bi5kZT6J
AJUDBRAwEHlOltEeLNENt4EBAbeIBACN3b6i9glcY2pOEz3nM5UbTxbH8YUK049Z
JoRU4y8QpOU8JTuovzz2UL8roo18kswT89GJMJiHnbgNtoMJQ55RbQQ9wbcYW0FP
KYL3vJNAGlvMQThf0jFC71HXp8+bLPPB4BKX8cdY1//5TeGefIhRGswXD5wyP+04

HhxMsKQhZokAlQIFEC+OlXErOCXSNjQ6+QEBJ0sEAIfjVech+rjpRYHim7jHZ6VJ
lexMTtEq1FEbANmDgsVNnpQefi0cDVxZrLkr4wCA2o+DP6e+j2Do3ytEJ9d1ZVXF
1RvGtSEO2LV69MGVLRe6bN8Y6Y8plxbPOfnmO2TR1tMbN2S4Zp7Mp9ZKUST6S4nu
LfA/tpdrHUKUDTfG1ARviQCVAgUQLx6dtQ9HL1s0103BAQGXWgP8CTGlMLUy5hoP
aZsN6zUxrhrTK49m8IfqRA2YZG31pXW9CCRobw7tU213i+AlkbGZJVE2hJVKdHOD
NR8G1iULMTh6maU5vO+dwtfB7QKSEtQ/oSx8kNBH8vgNwVkP9ouERy/GGg64G8Ic
XTjZiFUwTlUNTPu5Z911UFVmfCVr/rCJAFUCBRAu05iDbolZnIsnOJEBAZm8AfwM
UTwMcodf4V6ClwI0MI5jgQxEYbpyaM34PMdBncDhMIK/5DiyKy63QlspCzkId0+Z
BIiXeCvxz2GhDZVy/H58iQCVAgUQLu0ApkBgnhXGkfElAQG2SQQAkvZ2BL0aKLj0
8xta14hAKPAfq5ExlrBiv3PLR4JQFPenG6xu81Q3YOjarwxJydTEs6jklXAg+GCM
CoR4Mi4i83mFo5dHU4O4toUz2vUOTIvMjzG2ywKJy5+cg86EYpYH7E86bt6jmsmu
KqVAK9oM1wG2J/Yeu74ZqlpyyXUCnJ6JAJUCBRAu63mM6jqlORvSlCUBAcQZBADF
d5R2kJNQuzz4S6zrM8BdGcAEo4KGliJbQrRi96eg0pFCQWyga7/U8Vxn5da5TPmA
NoePiuRWJa//JA47IhFifGnt4pcB9xiIyk0QlTHmrZz8FfJMz2yXwhVhzgoACUnE
cEwxNvXJ9Pezd4udBRL3YNolnrCnU0Fz+J4FUwBRyIkAlQIFEC7lEYVVypIRmVSmQ
GQEBTqQEAOELmjXSMM7W2s9iTvnUbMlfA7JYeDmHqIK7bOOgEeyvzwGPdk41EK+b
cAsJnvXMDbTBh9IAIRp8QqzdW4B1HMl5dQ6o7k1RZQ3YAGKKA+yhGJUsFqLghMVT
KG48D6xs7bZfI/d4cAgochvGFEHddE1V+Y0vnt07KxnxTXraniP/tC9UaG9tYXMg
U2NodWVyZ2VyIDxqaG9ib3NAZXJjadG9zdGguZHNzZC5zdWIub3JnPokAlQMFEDAQ
eeyW0R4s0Q23gQEBFn0D/2WV9FlFIL6nCPuBNO6CODagb+lWlb/uQtvopnr2KXfB
c9GnbT5FypOdGcJvqICNx/7Cn4fxiBYbVDCGAOtOaaD/mclqj3McrCxECyhnbkGD
6GtMtqsISwTJk2GfqChuy5H+hZia8LlWPIv4lC5c+vlCPeggaKC5zrRpHrGYJ6uG
=yyg2
-----END PGP PUBLIC KEY BLOCK-----