

0164cf98-0

COLLABORATORS

	<i>TITLE :</i> 0164cf98-0		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	0164cf98-0	1
1.1	No title	1
1.2	Overview	2
1.3	Features	2
1.4	Why?	3
1.5	Requirements	3
1.6	Installation	3
1.7	Copyright	3
1.8	Disclaimer	3
1.9	Instructions	4
1.10	Editing	4
1.11	File Structure	4
1.12	The Header	5
1.13	The Note Information	5
1.14	Commands	6
1.15	CRPTRK11	6
1.16	author	7
1.17	title	7
1.18	comment	7
1.19	tempo	8
1.20	inst	8
1.21	module	8
1.22	XXXXXX	9
1.23	-----	9
1.24	CRPTRK10	9
1.25	loop	9
1.26	Notes	10
1.27	Med2Crp	10
1.28	Examples	11
1.29	History	11
1.30	Future	12
1.31	About the Author	13
1.32	Bugs	13

Chapter 1

0164cf98-0

1.1 No title

CrapTrack

v1.1

Copyright © 2002 DAS Software

Overview~~~~

What is it?

Features~~~~

What can CrapTrack do?

Why?~~~~~

Why use it?

Requirements

What is needed to run it?

Installation

How to install to a hard drive.

Copyright~~~

Legal information.

Disclaimer~~

More legal information.

Instructions

How to use it.

Med2Crp~~~~~

OctaMED to CrapTrack converter.

Examples~~~~

About the example modules included.

History~~~~~

How I was made.

Future~~~~~

What is to come?

Author~~~~~

How to contact the author.

Bugs~~~~~

Problems with CrapTrack.

1.2 Overview

The Amiga has many quality tracker programs, from the venerable Sound Tracker to the almighty OctaMED Sound Studio. However, if you want to use a really basic tracker, you have to turn to other platforms. The Atari ST has a wealth of dreadful Amiga tracker imitations, for example, whilst the PC has an inexplicably dire version of OctaMED Sound Studio.

That is, until now.

CrapTrack fills this vast hole in the market. It is the most basic tracker program available for the Amiga. And that includes the tracker supplied in the GVP DSS8+ sampler package.

Modules are edited using a text editor. It does not support any fancy effects. CrapTrack simply plays back four channels of note information.

1.3 Features

CrapTrack has various features. Or anti-features. These are:

- No Tooltypes.
 - No AHI support.
 - No soundcard support.
 - No sample mixing.
 - Load modules via ASL requester.
 - CLI interface.
 - WB argument parsing.
 - No support for fonts other than supplied XEN.font.
 - Only tracker to use a text editor as the module editor.
 - Module is shown on the screen whilst it plays back.
 - Modules can contain comments.
 - Accessing Intuition (ie. dragging a window, using the drop-down menus) freezes the module player.
 - OctaMED to CrapTrack conversion utility supplied.
-

1.4 Why?

CrapTrack is the ultimate back-to-basics tracker. Writing a good module with CrapTrack requires musical ability and a gift for melody, not a technical gift for knowing exactly when to use a C64-style chord warble or an AREXX script. It is also a test of patience.

Alternatively, it is just crap, but in a funky, skillo sort of way.

1.5 Requirements

CrapTrack requires:

- A real Amiga, or an emulator that supports the Amiga's custom chipsets.
- Workbench 2+
- A text editor.

1.6 Installation

To install CrapTrack, simply copy the CrapTrack directory to somewhere on your hard drive, then copy the contents of the Fonts/ directory to FONTS:

1.7 Copyright

CrapTrack is Copyright © DAS Software 2002. It is freeware, and ↔
as such is
freely distributable as long as all files in the archive remain intact.

CrapTrack can be included on magazine coverdisks/cover CDs, but I would like to be notified. I would prefer to receive a free copy of the magazine, or just the CD, but this is simply a curtesy and is not strictly necessary. Disk mags are also free to include CrapTrack if they wish, as are websites, etc.

Modules written using CrapTrack are of course freely distributable. The CrapTrack executable can be included with such modules without the rest of the archive. This is the only instance when the archive can be split and distributed separately.

If you are crazy enough to write a module using CrapTrack, please
email
me and
let me know about it!

1.8 Disclaimer

CrapTrack has been tested thoroughly using an A1200T 060/50MHz, an A1200 030/40MHz and WinUAE. However, you use CrapTrack at your own risk. I cannot be held responsible for:

- Damage to hardware
- Damage to software
- Damage to users
- Damage in general
- User stupidity
- Anything at all, in fact.

Remember: At your own risk. You have been warned.

1.9 Instructions

Instructions

Creating a CrapTrack module is fairly easy. I recommend looking at the included examples in the "Modules" directory to see how a module should be written.

Editing~~~~~

File~Structure

Commands~~~~~

Notes~~~~~

1.10 Editing

To edit a CrapTrack module, just load up your favourite text editor and begin. Note that CrapTrack expects Amiga-style line returns, not PC-style line and carriage returns.

1.11 File Structure

A CrapTrack module is divided up into two parts. These are the " ↔ header" and the "note information".

The~Header~~~~~

The~Note~Information

1.12 The Header

A typical header looks like this:

```
CRPTRK11

author Ant
title Dragon's Lair
comment Med2Trk

inst 00 samples/sample1
inst 01 samples/sample2
```

Of these, only two commands are actually necessary. These are the first command, "

```
CRPTRK11
", and the "
inst
" command. The others, "
author
", "
title
"
```

and "

```
comment
", just allow the module to report some information about its
```

creator.

1.13 The Note Information

The second part of the file is begun with the "module" command. After this comes the note information itself. Thus:

```
module
D-2-00:B-1-01:B-2-02:G#2-02
```

The "module" command separates the header from the footer. The second line above is the actual module data itself. This is structure thus:

```
Note-Octave-Sample
```

So, to play back sample number 00, using the second available octave at pitch "A", one uses the command:

```
A-2-00
```

For "A sharp", this is modified to:

```
A#2-00
```

As the Amiga has four channels of audio, each line must give instructions for every channel. Channel information is separated using a colon, ":". So:


```
A#2-00:C-2-01:C-3-02:C-1-03
```

If you do not want to play another sound, and wish the current sample to continue playing, simply use the command "-----" instead of specifying a note:

```
A#2-00:C-2-01:-----:C-1-03
```

If you want the current sound to stop playing, use the command "XXXXXX":

```
A#2-00:C-2-01:XXXXXX:C-1-03
```

Example modules are included in the "Modules" directory.

1.14 Commands

This is a list of all of the available commands.

Header commands:

```
CRPTRK11
```

```
author~~
```

```
title~~~
```

```
comment~
```

```
tempo~~~
```

```
inst~~~~
```

Note information commands:

```
module~~
```

```
XXXXXX~~
```

```
-----~~
```

Obsolete commands:

```
CRPTRK10
```

```
loop~~~~
```

1.15 CRPTRK11

CRPTRK11

Syntax: CRPTRK11

This command identifies the file as a CrapTrack module. It must be placed at the very start of a module.

1.16 author

author

Syntax: author <author name>

This is an optional command that displays the author's name in the CrapTrack window. The name can be up to 18 characters long.

Example:

```
author Ant
```

1.17 title

title

Syntax: title <module name>

This is an optional command that displays the module's name in the CrapTrack window. The name can be up to 18 characters long.

Example:

```
title Song40
```

1.18 comment

comment

Syntax: comment <text>

"Comment" is an optional command that displays a comment in the CrapTrack window. The comment can be up to 18 characters long.

Example:

```
comment This is crap!
```

1.19 tempo

tempo

Syntax: tempo <value>

This sets the tempo of the module, in vertical blanks. A tempo of 1 will play a new note every vertical blank, whereas a value of 10 will play a new note every 10 vertical blanks. In essence, the higher the number, the slower the song.

The default value is 5.

Example:

```
tempo 4
```

1.20 inst

inst

Syntax: inst <sample number> <path>

This command should be placed in the header of a module. It loads an IFF-8SVX sample into the module when it is played back. Sample number must be a two-digit figure. If the number only has one digit (ie. it is less than 10) then use the format 00, 01, 02, and so on.

Path is the complete name and path of the sample, relative to the location of the module.

Example:

```
inst 01 /samples/sample1
```

1.21 module

module

Syntax: module

This separates the header from the note information. It must come immediately before the note information, or the module will not play.

Example:

```
module
D-2-01:D-2-02:-----:XXXXXX
-----:-----:-----:-----
```

1.22 XXXXXX

XXXXXX

Syntax: XXXXXX

This is used in place of a note, and stops the current sample playing.

Example:

D-2-01:XXXXXX:D-2-01:D-2-01

1.23 -----

Syntax: -----

This is effectively a "gap". CrapTrack will do nothing upon meeting this command. Samples currently playing will continue to play.

Example:

D-2-01:-----:D-2-01:D-2-01

1.24 CRPTRK10

CRPTRK10

Syntax: CRPTRK10

This command is obsolete from Crap Track v1.1 onwards. It has been replaced with

CRPTRK11

.

This command identifies a file as a Crap Track v1.0 module. It should only be used in modules that conform to Crap Track v1.0 standards. There is no advantage to be gained in using this command in preference to

CRPTRK11

. If

used, it must be placed at the start of the file.

1.25 loop

loop

Syntax: loop <sample number> <path>

This command is obsolete from Crap Track v1.1 onwards. Samples that contain loop information are now automatically looped upon playback. However, the command is still present, to maintain backwards compatibility with Crap Track v1.0.

This command functions identically to the "`inst`" command, except samples loaded using it will be looped when played back. The loop information is read from the IFF-8SVX sample itself.

Example:

```
loop 01 samples/sample1
```

1.26 Notes

- Samples must be in IFF-8SVX format, or CrapTrack will probably crash.
- CrapTrack will play notes between the range C-1 and B-3.
- The number of samples is limited to 30.
- Inserting line returns inside the note information section of the module, in order to split the module into "blocks", is permissible.

1.27 Med2Crp

A slightly easier method of creating CrapTrack modules is to write them using OctaMED, then convert them to CrapTrack format using the Med2Crp utility. In order to achieve this, the modules must only use 4 channels of sound. Also note that these instructions pertain specifically to OctaMED Sound Studio, though I am fairly sure that at least OctaMED Pro 6, and possibly Pro 5, work in the same way.

Step-by-step instructions:

1. Load the module for conversion into OctaMED.
2. Select "Print" from the "Project" menu.
3. Click on the "All Blocks" button.
4. Select an output file.
5. "Print" the document to the file.
6. Save all instruments from the module as IFF-8SVX samples into a suitable directory.
7. Load the Med2Crp utility.
8. Click the "Input" button.
9. Select the file you saved using OctaMED.
10. Click the "Output" button.
11. Select an output file to save.

The module is now converted into CrapTrack format. To be able to play the module, all you need to do next is insert the necessary "inst" commands into its header using a text editor. After you have done this, you can load the module into the CrapTrack program and play it back.

Note that all data except notes, octaves and instrument numbers, is discarded.

One limitation of the Med2Crap utility is that it does not sequence the blocks into the order that OctaMED uses. Imagine a song which uses blocks in the following order:

```
00
01
00
01
02
```

The Med2Crap utility will just re-sequence this as:

```
00
01
02
```

In order to circumvent this limitation, you must either manually change the sequence of the blocks in the resulting CrapTrack module (all blocks have an empty line between them), or extrapolate the playing sequence within OctaMED before beginning the conversion.

1.28 Examples

There are three example modules included in the "Modules" drawer. These are:

Example.crp

This was labouriously typed by hand into "Ed".

DragonsLair.crp

This is the same module, but converted from the OctaMED original.

Song43.crp

This was also converted from an OctaMED module.

Song75.crp

As was this.

1.29 History

The first version of CrapTrack was written in approximately 2 days, minus a few hours for:

- Sleeping
 - Eating
 - Etc
-

It was partially based on a program I wrote back in 1995, designed to play the "Alex Kidd in Miracle World" titlescreen tune using nothing but Blitz code.

The Med2Crp utility also took just a couple of days to make.

CrapTrack was created using an A1200T 060/50MHz, with all the trimmings.

Release details

v1.1 - 27.6.02

- Added automatic looped sample detection, making the "loop" command obsolete.
- Changed CRPTRK header command to reflect new version.
- Fixed line numbers in scrolling display - did not reset after stop was pressed.

v1.0

- First release

1.30 Future

There are only a couple of features left to add to CrapTrack:

- Block labelling, leading to...
- ...block sequencing.

These might be added at some point. Look for more exciting updates on the Aminet, or on my

website

.

Other things that could be added, but won't, are:

- Main volume control
- Instrument volume control
- Note volume control
- Mid-song tempo changes
- AREXX port
- AppIcon
- AppWindow
- Font detection
- Output to sample
- Skinnable interface
- Realtime effects
- Channel mixing
- Soundcard support
- AHI support
- And so on.

CrapTrack is a back-to-basics tracker. The more features it possesses, the less crap it becomes.

1.31 About the Author

I can be contacted at:

spam_mail250@yahoo.com

Alternatively, have a look at my website:

<http://website.lineone.net/~greenpizza>

1.32 Bugs

Crap Track's window does not dynamically re-size itself to accommodate changes to the GUI made by Visual Prefs. Though Crap Track is already lacking in aesthetics somewhat, it should not be a problem unless you have used such odd settings that the interface's buttons are invisible. In this case, switch to a more sensible GUI design.

As far as I am aware, there are no other bugs in CrapTrack.