Advanced Audio Visualization Using ThMAD

Creating Amazing Graphics with Open Source Software

Peter Späth

Apress®

Advanced Audio Visualization Using ThMAD

Creating Amazing Graphics with Open Source Software

Peter Späth

Apress®

Advanced Audio Visualization Using ThMAD: Creating Amazing Graphics with Open Source Software

Peter Späth
Leipzig, Germany

ISBN-13 (pbk): 978-1-4842-3503-4 ISBN-13 (electronic): 978-1-4842-3504-1
https://doi.org/10.1007/978-1-4842-3504-1

Library of Congress Control Number: 2018938391

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Natalie Pao
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-3503-4. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Author

Peter Späth holds three degrees in physics: a master's from University of Texas at Austin (1994), a diplom from Universität Würzbug (1996), and a PhD from the Technische Universität Chemnitz (2000). He became an IT consultant in 2002. In 2016, he decided to concentrate on writing books on various topics, with the main focus on software development. Throughout his career he has always preferred to work with open source tools and favors operating system–independent platforms like Java. After using various audio visualization programs, he designed his own visualization suite, ThMAD, to overcome the deficiencies of important functionalities, quality documentation, and bug fixes of existing visualization suites.

About the Technical Reviewer

Massimo Nardone has more than 23 years of experience in security, web/mobile development, cloud computing, and IT architecture. His true IT passions are security and Android.

He currently works as the chief information security officer (CISO) for Cargotec Oyj and is a member of the ISACA Finland Chapter board. Over his long career, he has held many positions including project manager, software engineer, research engineer, chief security architect, information security manager, PCI/SCADA auditor, and senior lead IT security/cloud/SCADA architect. In addition, he has been a visiting lecturer and supervisor for exercises at the Networking Laboratory of the Helsinki University of Technology (Aalto University).

Massimo has a master's degree in computing science from the University of Salerno, and he holds four international patents (related to PKI, SIP, SAML, and proxies). Besides working on this book, Massimo has reviewed more than 40 IT books for different publishing companies and is the coauthor of *Pro Android Games* (Apress, 2015).

Introduction

This book is a sequel to *Audio Visualization Using ThMAD* and introduces advanced examples and features not covered in that book. It is *not* mandatory to have read that book if you already have considerable insight into the ThMAD program; if you are a beginner, then you will want to pick up a copy of *Audio Visualization Using ThMAD* because it contains a complete reference of the user interface and describes all the modules available in ThMAD.

For this book, the code is based on ThMAD version 1.1.0, and the associated OpenGL version as of the time of writing is 3.0.

Targeted Audience

This book is for artists with some IT background and for developers with artistic inclinations. Also, it is for readers of the first book, *Audio Visualization Using ThMAD*, who want to improve their proficiency in using ThMAD.

Installation

After you download ThMAD from `https://sourceforge.net/projects/thmad/` as a Debian package with the suffix `.deb`, you need to make sure the dependencies are fulfilled. A future version might do this for you

automatically, but for now you have to do it manually. ThMAD depends on the following packages; entries marked with an asterisk (*) are probably already installed on your Ubuntu Linux system:

- libglfw3 (≥ 3.1)
- *libc6 (≥ 2.17)
- *libfreetype6 (≥ 2.2.1)
- *libgcc1 (≥ 1:4.1.1)
- *libgl1-mesa-glx (≥ 11.2.0) or libgl1
- *libglew1.13 (≥ 1.13.0)
- *libglu1-mesa (≥ 9.0.0) or libglu1
- *libjpeg8 (≥ 8c)
- *libpng12-0 (≥ 1.2.13-4)
- *libpulse0 (≥ 0.99.1)
- *libstdc++6 (≥ 5.2)

To install these packages, log in as root in a terminal and enter this as one line:

```
apt-get install libc6 libfreetype6 libgcc1 libgl1-mesa-
glx libglew1.13 libglu1-mesa libjpeg8 libpng12-0 libpulse0
libstdc++6 libglfw3
```

If you downloaded ThMAD via a browser to the folder `Downloads` in your home directory, you will install it, still as root, via the following:

```
dpkg -i /home/[YOUR_USERNAME]/Downloads/thmad_1.0.0_amd64.deb
```

Make sure this is entered as one line and replace `[YOUR_USERNAME]` with your Linux username. If there is a newer version available, you can give it a try, but this book is for version 1.0.0.

All the files will end up in /opt/thmad. After that, please log off as root by pressing Ctrl+D. This is important so that subsequent actions do not mess with your system.

For your convenience, launchers are available; you can find them on your desktop after the following commands:

```
cp /opt/thmad/share/applications/thmad-artiste*.desktop ~/Desktop
cp /opt/thmad/share/applications/thmad-player*.desktop ~/Desktop
```

(Enter both in one line each.) While the installation main folder can be renamed, the launchers depend on the installation residing in /opt/thmad. You could, however, edit the launchers appropriately, if you think a different installation folder is a better option for you. To see whether everything works, use the launcher thmadartiste.desktop, or in the terminal enter /opt/thmad/thmad_artiste.

As another preparation step before actually using ThMAD Artiste, you might want to consider releasing the Alt key from the operating system. The default Ubuntu window manager, Unity, uses the Alt key to start the heads-up display (HUD), but ThMAD uses it for various GUI actions. To disable Ubuntu using the Alt key for the HUD or to change the key binding, go to the Keyboard section of Preferences, advance to the Shortcuts tab, and then go to the Launchers menu. Select the entry "Key to show the HUD" and press Backspace to disable it, or choose a new key or key combination to change the binding.

Upgrading ThMAD

If you have ThMAD version 1.0 running on your system, you can easily make all your files available to the newer version 1.1. All you have to do is copy your files from /home/[USER]/thmad/1.0.0 to /home/[USER]/thmad/1.1.0.

Conventions Used in This Book

Working with ThMAD is extensively coupled with using its modules, which are organized in a treelike structure. I'll usually refer to modules like `maths` ➤ `converters` ➤ `4float_to_float4`. If the module position inside the module tree is clear from the context, I'll shorten this to `4float_to_float4`.

State is the common notion for a rendering pipeline while constructing it. Finished *states* are also called *visuals*. References to sample states, including associated code provided with the installation and informational hints in general, are highlighted as follows:

Note This is a note. It might point you to a source file called `A-3.2.1_Visualization_basics_basic_samples_basic_2d_sample` in the `TheArtOfAudioVisualization` folder. By *folder* in this context I mean a folder showing up in the module lister or browser.

Tip You'll also find helpful tips like this.

Caution Important notes and pitfalls are marked like this.

Small code snippets appear directly in the text in monospaced font. Code and script snippets, as well as terminal input and output, usually show up as blocks in monospaced font like here:

```
apt-get install libc6 \libfreetype6 libgcc1 \libpulse0
libstdc++6 libglfw3
```

If a longer line does not fit onto the book page's width, a trailing ┐ at the end of each line of code signifies that the ┐ must be removed and the subsequent line break discarded. For example, the following:

```
echo "cmd [...] rectangle ┐
abc [...]"
```

should be entered as follows:

```
echo "cmd [...] rectangle abc [...]"
```

In many places, an asterisk (*) is used as a wildcard to denote any string. This frequently is used to refer to all the files inside a folder or to file name patterns.

Upon the first startup, ThMAD Artiste creates a data folder for all your states and visualizations at `/home/[USER]/.local/share/thmad` and creates a symbolic link at `/home/[USER]/thmad` pointing to the states and visualizations. If I'm referring to the data folder in this book, I will provide the link location.

How to Read This Book

This book should be read sequentially. To start, Chapter 1 serves as a concise operating manual for the different parts of the suite you are going to use, and Chapter 2 delves into meshes and particle systems in more depth.

Chapter 3 handles the way advanced timing issues can be addressed in ThMAD, allowing for a kind of video workstation view of ThMAD. Chapter 4 introduces how to use shaders for visualization purposes, as well as the way ThMAD handles them. Chapters 5 and 6 contain a collection of independent tutorials that you can work through in any order, with Chapter 6 concentrating on shader constructs. Chapters 7 and 8 talk about incorporating ThMAD in a JACK or ALSA sound server setup. ThMAD can be controlled from the outside, allowing you to bypass the Artiste user interface, which is described in Chapter 9. Chapter 10 explains advanced configuration issues.

CHAPTER 1

Program Operation

This chapter describes how the two main programs of the ThMAD software suite, Artiste and Player, can be started and stopped. It also describes all the possible options for controlling the programs.

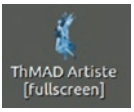
ThMAD Artiste Operation

ThMAD Artiste is the program for creating visualization sketches, called *states*, and viewing them in preview mode.

Starting and Using Different Modes

If you followed the installation instructions in the book's introduction, you will find two launchers on your desktop.

Start Artiste in windowed mode



Start Artiste in full-screen mode

Clicking the buttons takes you to `/opt/thmad/thmad_artiste` (for windowed mode) and `/opt/thmad/thmad_artiste -f` (for full-screen mode).

P. Späth, *Advanced Audio Visualization Using ThMAD*,
https://doi.org/10.1007/978-1-4842-3504-1_1

If you started ThMAD Artiste from inside a terminal, the full set of program options is available. For details about all the options, see Table 1-1.

Here is how you set individual options:

```
/opt/thmad/thmad_artiste [option1 option2 ...]
```

Table 1-1. *Artiste Program Options*

Option	Description
<none>	Starts Artiste in windowed mode. This option shows the canvas for creating sketches, a small preview window, and a module list.
-help	Shows help and immediately quits the program.
-h	Same as -help.
-sm	Prints all detected monitors and immediately quits the program. You can use the output to specify a monitor number for the -m option.
-m mon	Uses the monitor number for full-screen mode. This has no effect if not used with the -f option.
-f	Starts Artiste in full-screen mode. It is not possible to switch to full-screen mode from inside the program. You can exit this mode by pressing the Esc key. This option can be used in conjunction with the -ff and -fn options.
-ff	Starts Artiste in full-window mode. The graphics output will use the complete window space. You can later switch back to the non-full-window standard mode by pressing Ctrl+F.

(*continued*)

Table 1-1. (*continued*)

Option	Description
`-fn`	If in full-window mode, suppresses the info text in the header area.
`-s 1024x860`	Sets the window size in windowed mode. `1024x860` is just an example; see the output of the `-sm` option for possible values.
`-p 200x100`	Sets the window position; `200x100` is just an example.
`-novsync`	Experimental; disables using double buffering.
`-gl_debug`	Experimental; activates special OpenGL debugging feature.
`-port 3267`	Starts a TCP/IP port where commands to control ThMAD from outside may be sent to. `3267` is just an example.
`-sound_type_alsa`	Directly uses the ALSA API instead of PulseAudio.
`-sound_type_jack`	Use a JACK sound server endpoint to connect.
`-snd_rtaudio_ device=5`	If using ALSA or JACK, specifies the sound device to use. Sound devices get listed upon startup, but the `audio_visualization_listener` module must be present.

By default, full-window mode shows some status information in the header area. It can be disabled or enabled by pressing Alt+T. Or you can start Artiste with the `-fn` option to disable the status information from the beginning.

There is also a performance mode, which presents an overlay of the state creation canvas and the graphics output. To enable it, start in full-window mode or switch to full-window mode (Ctrl+F) and then press Alt+F. You can leave performance mode by pressing Alt+F again. Also, when in performance mode, you can toggle the visibility of the header info lines by pressing Alt+T.

To leave the program in any mode, press the Esc key, or, if available, click the Close button of the window or use the main context window.

Stopping ThMAD Artiste

You can stop ThMAD Artiste via any of the following:

- Right-click an empty spot of the canvas and select Exit. ThMAD detects if you have saved your changes; if this is not the case, it will ask you whether you really want to exit.
- Press the Esc key. ThMAD will tell you if there are unsaved changes.

You can also use a module called `system` → `shutdown` to shut down from inside a rendering pipeline. You can place it on the canvas and connect it to the screen module `screen0`. As soon as the module's input exceeds 1.0, the program will shut down.

Artiste Files

ThMAD Artiste will look for its files in the following folder:

`/home/[USER]/.local/share/thmad/[VERSION]`

and in these subfolders:

- `/states`: From here the states are loaded, and this is where they get saved.

- `/resources`: Here Artiste will look for or save resources such as data files and images.
- `/visuals`: This is used only when exporting finished states as visuals.
- `/prods`: This is used only when exporting finished states as prods.
- `/faders`: This is used only when saving faders.
- `/macros`: When macros get saved, they will go here.

During installation, a link to the folder `/home/[USER]/.local/share/thmad` gets created at `/home/[USER]/thmad` for convenience.

The single configuration file used by Artiste for defining some settings is located here:

```
/home/[USER]/thmad/[VERSION]/data/thmad.conf
```

Chapter 9 explains more about the configuration.

ThMAD Player Operation

With ThMAD Player, you can play visuals, which are exported states from Artiste. This happens when you invoke the Compile ➤ Music Visual command from Artiste's main context pop-up.

Starting and Using Different Modes

In default operation mode, ThMAD Player will recursively register all the visuals it finds inside the user's data folder and play them one by one. If you have used ThMAD's predecessor VSXu, where ThMAD Player by default looks in the installation folder, you should be aware of this difference.

Also, contrary to Artiste's operation, Player knows how to handle *faders,* which introduce a transition between visuals when it comes to switching from one to another. Faders are also created inside Artiste and from there exported via Compile ➤ Music Visual Fader. They then end up inside the `faders` folder.

Player will, however, not see your exports automatically since the data spaces for Player and Artiste are kept separate. To make exported states available to ThMAD Player, you have to copy the visuals and possibly the faders from the following locations:

```
/home/[USER]/thmad/ [VERSION]/data/visuals

/home/[USER]/thmad/ [VERSION]/data/faders
```

and put them in the following locations:

```
/home/[USER]/thmad/ [VERSION]/data/player_visuals

/home/[USER]/thmad/ [VERSION]/data/player_faders
```

Alternatively, you can copy them to some other place and tell Player via a startup option where to find them (using the `-path` flag).

If not using a launcher but instead the terminal to start Player, you can use all the available options, as shown in Table 1-2.

Here is how to set certain options:

```
/opt/thmad/thmad_player [option1 option2 ...]
```

Table 1-2. *Player Program Options*

Option	Description

<none>	Starts Player in windowed mode. It also recursively loads all visuals from the following path:
	`/home/[USER]/thmad/`
	`[VERSION]/data/ player_visuals`
	and uses all faders found in this folder:
	`/home/[USER]/thmad/ [VERSION]/data/ player_faders`
	Visuals and faders are played in random order, each running for 30 seconds. Note that `/home/[USER]/thmad` is a symbolic link to the following location:
	`/home/[USER]/.local/ share/thmad`

`-help`	Shows help and immediately quits the program.
`-h`	Same as `-help`.

`-path PATH`	Does not load the visuals from the local user data path; see the <none> options. Instead, it loads all the visuals from the following path:
	`PATH/player_visuals`
	and uses all the faders found in this folder:
	`PATH/player_faders`

`-dr`	Disables the randomizer. Player will then not automatically cycle through the available visuals. Still, the visual chosen will be a random one.

(*continued*)

Table 1-2. (*continued*)

Option	Description
`-rb 20`	Sets the number of seconds to wait before changing to the next visual, if the randomizer is *not* disabled. If this option is not given, the value defaults to 30 seconds.
`-rr 10`	Randomizes the randomizer, if not disabled. A random visual duration will be chosen between the base number from the `-rb` option and the `-rb` number plus the `-rr` value. In this example, it's between 20 and 30 seconds. If this option is not given, the value defaults to 0 seconds.
`-f`	Starts in full-screen mode.
`-sm`	Lists available monitors and monitor modes.
`-m 2`	If in full-screen mode, uses monitor 2 in this example.
`-fm`	Lists available video modes for full-screen mode. Depends on the monitor chosen (see the `-m` option).
`-p 300x200`	If in windowed mode, sets the window position to (300;200) in this example.
`-s 640x480`	If in windowed mode, specifies the window size. 640×480 is only an example; choose any size you like. If in full-screen mode, this may be used to set the resolution. ThMAD then tries to find the best possible match. See the `-fm` option for a list of available video modes. If this is not given and the full-screen mode and possibly some monitor are requested, the video mode will automatically be chosen based on your current settings. Letting the system choose is the preferable way.
`-no`	Specifies no splash screen and overlay. This means Player will start immediately with the first visual, and it will not print a visual's name at its beginning.

(*continued*)

Table 1-2. (*continued*)

Option	Description
`-lv`	Lists visuals seen by the Player. This depends on the `-path` option if chosen.
`-lf`	Lists faders seen by the player. This depends on the `-path` option if chosen.
`-port 3267`	Starts a TCP/IP port where commands to control ThMAD from outside may be sent to. `3267` is just an example. The details of the protocol are not part of this book.
`-sound_type_alsa`	Directly uses the ALSA API instead of PulseAudio.
`-sound_type_jack`	Uses a JACK sound server endpoint to connect.
`-snd_rtaudio_device=5`	If using ALSA or JACK, specifies the sound device to use. Sound devices get listed upon startup, but the `audio_visualization_listener` module must be present.
`-schedule <S>`	Specifies a schedule; see the next section for more information.

Unlike ThMAD Artiste, in ThMAD Player the visual will immediately cover the whole window or screen, and there is no context menu for the player. You can, however, press F1 to get some basic on-screen help.

Note If you request a certain resolution in full-screen mode, it may cause the ThMAD program to terminate and show your desktop in that new resolution. You may have to manually revert the resolution setting or restart your desktop if you want to switch back to the resolution you are accustomed to.

Scheduling the Player

You can create a storyboard by using the `-schedule` switch, which was introduced earlier in the chapter. The syntax is as follows:

```
-schedule ind1:s1,ind2:s2,ind3:s3,...
```

Here, `indN` is an index in the alphabetically sorted Player files, and `sN` is the number of seconds as a floating-point value. The list is zero-based and the same as seen by the outcome of the `-lv` switch.

For example, if inside the visuals directory you have three visuals (called A_visual1, B_visual2, and C_visual3) and one storyboard that says "play B_visual2 for 15.4 seconds, then A_visual1 for 10 seconds, then C_visual3 for 102.4 seconds, and then again B_visual for 45.1 seconds," you'd write the following:

```
/opt/thmad/thmad_player -schedule 1:15.4,0:10.0,2:102.4,1:45.1
```

Here, A_visual1 has index 0, B_visual2 has index 1, and C_visual3 has index 2 in the list. You can also write it as one line, in which case you'd omit the backslash.

Stopping the Player

You can stop ThMAD Player with either of these actions:

- Press the Esc key with the focus on the ThMAD Player window. In full-screen mode, no focus is needed.
- If while constructing the state you placed the module `system` → `shutdown` on the canvas and connected it to `screen0`, as soon as the module's input exceeds 1.0, the program shuts down.