

RetinaANIM

COLLABORATORS

	<i>TITLE :</i> RetinaANIM		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	RetinaANIM	1
1.1	RetinaANIM Documentation	1
1.2	Introduction to RetinaANIM	1
1.3	How to Install RetinaANIM	2
1.4	About the Author	2
1.5	How to use RetinaANIM	2
1.6	What might RetinaANIM do in the future?	4
1.7	Known problems with RetinaANIM	6
1.8	Things to watch out for with RetinaANIM	6
1.9	RetinaANIM release history	6
1.10	Copyright notices for various companies	7

Chapter 1

RetinaANIM

1.1 RetinaANIM Documentation

RetinaANIM Version 39.5:

RetinaANIM - Animation utility for the Macromedias Retina Z3.

[Introduction](#)

[Installation](#)

[About Author](#)

[Usage](#)

[Future](#)

[Known Bugs](#)

[Warnings](#)

[History](#)

[Legal](#)

1.2 Introduction to RetinaANIM

Introduction:

This is a program that will display animations on the Retina Z3 graphics card. It will not work on the Retina Z2.

When completed it will play AVIs, QuickTime's and IFF ANIMs. It may even play MPEG 1 (maybe MPEG 2) if I get really enthusiastic. Currently however it supports the following:

* AVI files

Video Codecs:

* Microsoft CRAM.

* Radius Cinepak.

Audio Codecs:

* WAV PCM.

* WAV ADPCM.

RetinaANIM requires:

* AmigaDOS 3.0 (version 39) or greater.

* 68030 minimum.

* Retina Z3 graphics card. It will not work with the RetinaZ2 (yet).

If RetinaANIM does not support an AVI you have, or you have any really obscure AVIs, please e-mail me at this [address](#) so I can arrange for it to be sent to me for testing/development etc.

1.3 How to Install RetinaANIM

Installation:

Installation is easy!!

Just copy RetinaANIM anywhere, and use it. If you have a 68040, then use RetinaANIM040 as it is optimised for the '040, and uses 68881 instead of standard math - for the few places it is used. Because of the '040 instruction scheduling it should be faster. This speed difference may be more noticeable with a superscalar processor like the 68060.

Then you should place the RetinaANIM.guide file somewhere (if you keep ALL documentation).

RetinaANIM is so easy to use, this file is probably not very important.

1.4 About the Author

About Author:

RetinaANIM was written over a period of four (4) months by myself, Stuart MacKinnon.

If anyone has an idea for a program and they don't know how to do it themselves (as they may not be a programmer), I would be glad to help.

I can be contacted via Internet (e-mail):

stuartm@zip.com.au

Or by snail-mail:

6/5 Abbey Street

Randwick NSW

Australia 2031

1.5 How to use RetinaANIM

Usage:

Firstly **install** RetinaANIM. Technically you don't have to install it just to test it out. You could just enter its name from the Shell.

RetinaANIM can only be started from CLI/Shell. If you run it from the Workbench it will inform you of this.

The option template is:

FILE/A/M,BUFSIZE/N,VOL/N,STARTDELAY/N,ENDDELAY/N,FRAMETIME/N,PRI/N,AUDPRI/N,
NOAUDIO/S,NOVIDEO/S,NODBUF/S,SKIP/S,VERBOSE/S

FILE/A/M: File name of the animation to play. This is a multiple result argument, meaning that you can supply more than one file name on the command line. It also supports wild cards for this option.

BUFSIZE/N: Size of the disk I/O buffer. Disk I/O is double buffered, so two (2) buffers of this size will be allocated. This option is in bytes. Default is 1048576 (1 Mbyte). If you find that the animation slows down a lot during disk I/O, then it may be better to have a smaller value here so that disk I/O is spread more evenly across the whole animation giving much more consistent playback.

VOL/N: The volume of the audio (0 to 64). Default is 64.

STARTDELAY/N: Number of seconds to wait after opening the screen before the animation should be started. This is so that if you have a monitor that takes a second or two to re-sync to the new screen mode, you will be able to see the whole animation. Default is 0.

ENDDELAY/N: Number of seconds to wait on the last frame before closing the screen. Default is 0.

FRAMETIME/N: Time between each frame. This can be used to slightly adjust the frame time if audio seems to be slightly out of sync with video. The VERBOSE option will tell you what the frame time is normally. If you set the frame time too large the animation will play slowly and you will probably hear gaps in the audio. If the frame time is too small, then the audio will continue to play at normal speed, and it will therefore be out of sync with video.

PRI/N: This gives the priority of the video/audio decoding task.

Default is 15. This may seem high, but for smooth playback it should be higher than the filesystem task. There is reason for this high number. If the number is lower than the filesystem, then when the file is being read, the main process has very little CPU time to decode each frame.

However, when the main process is higher than the filesystem, there is enough time during the frame time wait for the filesystem to do its thing. The result is much more consistent play back rates. With the main task's priority lower than the filesystem, play back seems jerky especially when disk access is performed. 15 works well with my A4091, as well as the IDE controller in an A4000.

AUDPRI/N: This gives the priority of the audio playback task. Default is one higher than the video/audio decoding task.

NOAUDIO/S: This disables audio playback/decoding, so you can just see

video.

NOVIDEO/S: This disables video playback/decoding, so you can just hear audio. If you want to hear the audio, but the video decoding takes too long (and the audio then skips), use this option.

NODBUF/S: This disables double buffering of the display. In some cases it may speed up the AVI by a small amount. You will probably see the screen refresh however, and that looks ugly.

SKIP/S: Normally if a couple of frames of video take a bit long to decode, then the following frames will be played as fast as possible to attempt to catch up. Once video has caught up, video is decoded and the appropriate frame time is put back in. This option will cause frames to be completely skipped to catch up. It will only skip frames if it gets 2 frames or more behind. There are problems with this option. Some video codecs use delta information for each frame. That means that if a frame is skipped, then the frame buffer is not in the correct state for the next delta frame. What happens is that the display looks corrupted until the next key frame (or frame that contains a full screen of data) is displayed. A codec that repaints the whole screen every frame will not have this problem.

VERBOSE/S: This option will give information about the animation headers such as what video codec is used, what audio format was found, frame time, width and height of screen, sample rate of audio and number of channels, etc.

RetinaANIM will open an 8, 16 or 24 bit screen depending on what the animation requires. It will use the defaults configured in RetinaScreenMode.

1.6 What might RetinaANIM do in the future?

Things to do:

There are quite a few things to do to RetinaANIM. Those features are:

- * Allow for an environment variable that gives defaults for options such as VOL, QuickTime resource and data fork extensions (the options don't actually exist yet), VERBOSE, NODBUF, STARTDELAY, ENDDDELAY, PRI, AUDPRI, SKIP and BUFSIZE.
 - * Fix audio sync'ing problem. See [bugs](#) for more information.
 - * Proper scanning of an AVI's indx chunk, and using its results.
 - * Looping of AVIs (eventually all animations).
-

- * Do QuickTime support.
- * Do IFF ANIM support.
- * Do MPEG 1 (maybe even MPEG 2) support.
- * Make codecs separate executables/libraries.
- * Make it all into a library, such as movie.library, and therefore make it possible for other coders to access the AVI/QT reading (and perhaps writing) and decoding routines for their own applications. The library would take care of using the correct codec(s). When writing, the library would take care of producing the correct index chunks, offsets etc. This is something I would like to do, but it all takes time. The idea would be that an area of memory would be allocated by the application using the specifications that library gave about the movie, and then passed to the library. The library would then fill the area of memory with the decoded data, and the application would be told about it. The library would be told what type of buffer it is, such as chunky/planar/RG-B/RGBA/BGR/etc. Also there would be modulus dealt with etc. That way the library could place the data in a buffer, or direct to screen memory etc. The library would have similar functions for audio. These would be the low level functions. There would be higher level functions that would just play the movie with the correct frame delay etc. I would envisage that all the low level functions would exist to do things like parse any index/offset information, waiting the correct frame time, generally parsing the movie, etc. Then the application could call the higher level playing function, or do it itself with the low level functions. Lots of work to do to make it like this, but I think it would be worth it. For example, a datatype could be written that would use these functions.

Graphics Cards:

- * Standard Amiga.
- * EGS.
- * Retina-Z2.
- * Piccolo and SD64.
- * CyberVision 64.
- * Picasso II.
- * Merlin.

Video Codecs:

- * 00DC.
- * CRAM16 (Microsoft CRAM 16 bit).
- * RGB.
- * RGB32.
- * RLE8.
- * RLE4.
- * ULTI (IBM Ultimotion).
- * RT21 (Intel Indeo V2.1).
- * IV31 (Intel Indeo V3.1).
- * IV32 (Intel Indeo V3.2).

Audio Formats:

- * WAVALAW.
 - * WAVMULAW.
 - * WAVOKIADPCM.
 - * WAVDIGISTD.
 - * WAVDIGIFIX.
-

* IBMMULAW.

* IBMALAW.

* IBMADPCM.

The problem with providing support for the codecs is actually having some AVIs that contain data in these formats so I can test the codecs and optimise their performance. The other problem is that (in the case of Intel Indeo stuff) some codecs are proprietary and the companies who designed them don't like giving up data formats without lots of money for license fees, or NDAs etc.

IMPORTANT NOTE:

If you would like to have RetinaANIM play AVIs using the Intel Indeo(r) codec, then please send money to me. The Intel Indeo(r) porting package would cost me US \$5000.00 to obtain. I don't have that kind of money, but if others contributed, then I may be able to pay a much reduced amount.

Note: If you do send money, it would be used for the express purpose of obtaining the Indeo(r) porting package. If after a few months I have not been able to get enough money (along with my own) to obtain this package the funds will be returned to you.

Here is my [address](#) to send money to.

1.7 Known problems with RetinaANIM

Known Bugs:

Sometimes the audio gets out of sync with video. If for some reason your machine is too slow, and the video cannot be decoded fast enough, then the audio buffer will be played and the audio will stop until it gets more audio data. Most AVIs have a few frames worth of audio before any video data. That means that in most cases two adjacent parts of audio and video are not meant to be played at the same time, the audio is a few frame ahead of the video in the file, and is buffered. However if the video is too slow and the audio stops, when new audio is available it will not be sync'd with the video any more. I cannot think of an easy and clean way to get around this - well I can I am just saying this as an excuse for not doing it.

1.8 Things to watch out for with RetinaANIM

Warnings:

This will only work with the Retina Z3!!! It directly accesses Retina Z3 memory. For speed (it makes a huge difference), the code does not deal with the segmented architecture of the Retina Z2. Perhaps at a later time I will have two sets of code, one for the Z3, and the other for the Z2 - which will be slower. If you attempted to use it on a Retina Z2 you would probably see a few scanlines that would be correct (in fact 64K of data in the frame would seem valid). Then whatever is autoconfig'd after your Retina Z2 would be accessed. This could set off HD controllers, or anything else you may have plugged into your machine. You have been warned!!

With big sized frames, it chews up a lot of CPU time, but I guess you knew that already!

1.9 RetinaANIM release history

History:

RetinaANIM 39.5 (30.1.96)

Third AMINET release.

- Now does preliminary IFF ANIM parsing. Just feed it an ANIM file and use the VERBOSE option to see (same with QT).

RetinaANIM 39.4 (28.1.96)

- Now has a FILE/A/M option instead of just a FILE/A.

- Has wild card support for the FILE/A/M option.
- Fixed a bug where the async I/O routines would attempt to LOCATE_OBJECT relative to NULL lock, which would work if the full path to the AVI was given, otherwise it would fail, and the failure condition was not being handled properly. It now obtains the full path, and does LOCATE_OBJECT relative to NULL still, and now handles the failure condition correctly.

RetinaANIM 39.3 (24.1.96)

- Now prints more descriptive text for the INFO chunk sub chunks.

RetinaANIM 39.2 (21.1.96)

Second AMINET release.

- Fixed a bug where sound would overwrite the end of its audio buffer causing lots of enforcer hits or in most circumstances a nasty crash (even with enforcer running). This would only happen with some ADPCM AVIs. All fixed now.

RetinaANIM 39.1 (6.1.96)

Initial AMINET release.

1.10 Copyright notices for various companies

Copyright notices:

RetinaANIM, its documentation, and support material is © Copyright 1996 Stuart MacKinnon. All rights reserved.

This product is giftware. If you like it you could send me money, programs, books, CDs, AVI files, or anything else you care to send me. Here is my [address](#).

There are no warranties expressed or implied as to the performance of this product or the suitability of this product for a particular purpose. No responsibility is taken for any damage, lost revenue or time caused by the use of this product.

Cinepak is a trademark of Radius Inc.

Indeo is a trademark of Intel Corp.

Ultimotion is a trademark of International Business Machines Corp.

QuickTime is a trademark of Apple Computer Inc.

68030, 68040, 68060, 68881 are trademarks of Motorola Inc.

All other trademarks or registered trademarks are of their respective companies.
