

AMP

COLLABORATORS

	<i>TITLE :</i> AMP		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 27, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AMP	1
1.1	AmiDog's Movie Player - The Movie Player for your PPC Amiga!	1
1.2	AmiDog's Movie Player - Introduction	2
1.3	AmiDog's Movie Player - Requirements	2
1.4	AmiDog's Movie Player - Features	3
1.5	AmiDog's Movie Player - Installation	4
1.6	AmiDog's Movie Player - Usage	4
1.7	AmiDog's Movie Player - Control Panel	5
1.8	AmiDog's Movie Player - Speed	5
1.9	AmiDog's Movie Player - FAQ	6
1.10	AmiDog's Movie Player - FAQ - HAM Width	6
1.11	AmiDog's Movie Player - FAQ - HAM Quality	7
1.12	AmiDog's Movie Player - FAQ - HAM S-Lores	8
1.13	AmiDog's Movie Player - ShareWare	8
1.14	AmiDog's Movie Player - Disclaimer	8
1.15	AmiDog's Movie Player - Development	9
1.16	AmiDog's Movie Player - Bugs	9
1.17	AmiDog's Movie Player - Contact	9
1.18	AmiDog's Movie Player - Thanks!	9
1.19	AmiDog's Movie Player - Future	10
1.20	AmiDog's Movie Player - History	10

Chapter 1

AMP

1.1 AmiDog's Movie Player - The Movie Player for your PPC Amiga!

AmiDog's Movie Player v1.30, 991120

Copyright 1998-99

Mathias "AmiDog" Roslund

Introduction

- What is this?

Requirements

- What does it require?

Features

- What can it do?

Installation

- How to install?

Usage

- How do I use it?

Control Panel

- How do I use this?

Speed

- How fast is it?

FAQ

- Have someone asked this?

Shareware

- What does this mean?

Disclaimer

- Who's responsible?

Development

- How is it done?

Bugs

- Are there any known bugs?

History

- What's new?

Future

- What will be added?

Contact

- How to contact me?

Thanks!

- Which have helped me?

Visit the homepage at <http://www.amidog.com/amp/>

1.2 AmiDog's Movie Player - Introduction

This is a Movie Player for Amigas equipped with a PPC board.

I started writing this in november 1998 since there were no MPEG players for PPC which had good AGA support. And as you can see, the AGA support is probably one of the best in any movie player for the Amiga.

Already from the beginning I planned to add support for more media types than MPEG, but it's not until recently (v1.20) that I've had the time to do so.

This product is released as Shareware, which means that you can freely use it for a period of 30 days, but then you must send the author a small fee. There are also some restrictions in the unregistered version, it only support MPEG.

Now start the player and enjoy the speed of your PPC board!

1.3 AmiDog's Movie Player - Requirements

Hardware:

- * Amiga 1200/3000/4000 with a PowerPC board
- * 16 MB RAM
- * OS 3.0+

Software:

- * WarpUP v4.0
- * CGFX v3 or later for CGFX support

1.4 AmiDog's Movie Player - Features

It currently supports the following multimedia types:

- * MPEG1/2 video with sound (optional) and automatic frameskip.
- * FLI/FLC (most chunks supported, please report any problems).
- * AVI (look below for supported codecs).
- * QT (look below for supported codecs).

NOTE: The unregistered version only got MPEG support.

Supported encodings:

Video:

(AVI, QT)	Radius CinePak	(CVID)	8/16/24/32 bit
(AVI)	Microsoft Video 1	(MSVC)	8/16 bit
(AVI)	?	(YUY2)	16/24/32 bit
(AVI, QT)	JFIF JPEG	(JPEG)	8/24 bit
(QT)	Apple Video	(RPZA)	16 bit
(AVI)	Microsoft RLE8	(RLE8)	8 bit

Audio:

(AVI, QT)	PCM/PCMO	(Mono/Stereo)	8/16 bit
(QT)	IMA ADPCM	(Mono/Stereo)	16 bit
(QT)	TWOS	(Mono/Stereo)	8/16 bit
(QT)	MACE 3:1	(Mono/Stereo)	8 bit
(QT)	MACE 6:1	(Mono/Stereo)	8 bit
(AVI)	MS ADPCM	(Mono/Stereo)	4 bit (decodes to 16 bit)

NOTE1: Even though you actually can play 8bit FLI/FLC/AVI/QT files in any higher depth (15/16/24/32 bit), it's not recommended since it's slower than 8bit ←

NOTE2: I could easily add support for more codecs. If you've got an AVI/QT using a non supported, non Indeo codec, please tell me which codec it uses and I'll do my best to add support for it, but please don't e-mail any anims unless I ask you to. AVI/QTs on disk/Zip/CD is however welcome.

The features include:

- * Fast AGA support with 10 HAM modes.
- * CGFX support using optional direct videomem access.
- * Demuxing AsyncIOPPC routines (MPEG).
- * General AsyncIOPPC routines for FLI/FLC/AVI/QT etc.
- * Fast audio playback using audio.device.
- * Frameskip (automatic, MPEG only).

- * CLI support and GadTools MiniGUI.
- * Window support (CGFX) with optional speed hack (FAST!).
- * Overlay support (On BPPC, you MUST use BPPC fix!).

1.5 AmiDog's Movie Player - Installation

Just copy it to any place on your HD.

The keyfile you get when you register should either be placed in the same directory as the AMP executable or in the S: assign.

1.6 AmiDog's Movie Player - Usage

1. start shell.
2. stack 1000000 <ENTER>.
3. amp gui, amp "filename" or just amp (for help)<ENTER>.

NOTE: If AMP crashes, try to increase the stack!

Quit playing by pressing LMB or closing the window.

The following commandline options are supported:

```

REQUESTER/S      : use an ASL requester to get the screenmode (def: BestMode)
PAL/S            : use PAL screenmodes (def: BestMode)

GRAY/S          : Gray output (def: color)
HAM/S           : HAM output (def: color)

HAMDEPTH/N      : HAM depth (6 or 8 bitplanes, def: 8)
HAMWIDTH/N      : HAM width (1, 2 or 4 HAM pixels per RGB pixel, def: 2)
HAMHQ/S         : HAM high quality (def: normal quality)
HAMSLORES/S     : HAM super lores on (def: off)

GRAYDEPTH/N     : Gray depth (4, 6 or 8 bitplanes, def: 8)
FRAMERATE/N     : FrameRate (1 for maximum, or 5, 10, 15, ..., 60, def: FromStream ←
)
SOUND/S         : sound on (def: off)
LOWCOLOR/S      : 8bit CGFX (used for BestMode)
HIGHCOLOR/S     : 15/16bit CGFX (used for BestMode)
TRUECOLOR/S     : 24/32bit CGFX (used for BestMode)

WINDOW/S        : use a window on Workbench for playback (def: off)
SPEEDHACK/S     : speed hack on (def: off)
DEBUG/S         : use this if you have problems and send me the output (def: off)
VERBOSE/S       : gives additional information while/after playing (def: off)
OVERLAY/S       : enable overlay support (def: off)
* STEREO/S      : play and decode stereo audio (def: off)
* DUALMONO/S    : play mono audio on both left and right channel (def: off)

DEFAULT/S       : save current settings as defaults

```

NOTE: Options with the prefix * aren't working currently.

1.7 AmiDog's Movie Player - Control Panel

To be written... (when finished)

1.8 AmiDog's Movie Player - Speed

CGFX speed: (Measurements using v1.30, CGFX v4 using a BVision and a 240*176 MPEG ↔)

```
- Color Mode - AMP - IsisPPC -
* 8bit (Gray) : 71.1 fps. : N/A.
* 8bit (Color) : 47.4 fps. : 27.9 fps.
* 16bit (Color) : 44.9 fps. : 29.4 fps.
* 24bit (Color) : 40.8 fps. : 28.9 fps.
```

AGA speed : (Measurements using v1.30, AGA in PAL Lores and a 240*176 MPEG)

```
- Color Mode - AMP - IsisPPC -
* 4bit (Gray) : 58.0 fps. : N/A.
* 6bit (Gray) : 52.4 fps. : N/A.
* 8bit (Gray) : 46.7 fps. : N/A.
* 8bit (Color) : 34.5 fps. : 12.9 fps.
* HAM6 (1 Width): 23.3 fps. : N/A.
* HAM8 (1 Width): 22.1 fps. : N/A.
* HAM6 (2 Width): 31.9 fps. : N/A.
* HAM8 (2 Width): 26.5 fps. : N/A.
* HAM6 (4 Width): 22.8 fps. : N/A.
* HAM8 (4 Width): 12.6 fps. : N/A.
```

SuperLores speed: (Measurements using v1.30, AGA in PAL and a 160*120 MPEG)

```
- Color Mode - On - Off -
* HAM6 (2 Width): 58.2 fps. : 58.1 fps.
* HAM8 (2 Width): 52.0 fps. : 50.1 fps.
* HAM6 (4 Width): 44.9 fps. : 43.3 fps.
* HAM8 (4 Width): 36.1 fps. : 25.4 fps.
```

NOTE1: For the HAM modes, HAM Quality is Normal.

NOTE2: SuperLores PAL means a 640/320*128 screen, while a normal Lores is ↔ 640/320*256.

NOTE3: Of some reason, IsisPPC reports 163 frames the first time the 240*176 MPEG ↔ is played, but 161 frames all of the following times, strange!

NOTE4: AMP is 2.67 times faster than IsisPPC on AGA using 8bit color and PAL.

The 240*176 MPEG is a Wallace and Gromit MPEG played from RAM.
The 160*120 MPEG is a Tintin MPEG played from RAM.

All tests are performed using an A1200T 603e'200, 040'25, 128MB and OS3.1 from ↔
Workbench.

1.9 AmiDog's Movie Player - FAQ

What are these settings good for?

HAM width

HAM quality

HAM slores

1.10 AmiDog's Movie Player - FAQ - HAM Width

Back

To understand this you must first know the basics about how HAM ↔
modes work. I'm
not an expert on this, and just before christmas last year (1998), I didn't know
much at all. Anyway, here goes.

On a HAM screen a pixel is either one from the 16/64 colors in the palette or a
"finetuned" version of the pixel next to the left, that is, you can modify one
of the three color components Red, Green, or Blue (RGB). Two bits per pixel is
used to determine if "finetune" or palette color should be used. That is why a
HAM6 screen is 6bit but only gives you a 16 color palette (4bit).

This means that to get one specific RGB value, you must either make sure that it' ↔
s
one of the 16/64 colors in the palette, or you must use three (3) pixels on ↔
screen
to achieve the correct RGB values. This is why AMP supports different HAM widths.

As you might know, there are no 3-width screenmodes available, for example, there
are 320*256 (1-width), 640*256 (2-width) and 1280*256 (4-width) for PAL. So to be
able to get the right RGB value, you must use a screen which is four (4) times as
wide as it is high. Since you then will get the right RGB value, these screens ↔
are
often callen 12bit/18bit since you get 4/6bit per RGB component and there are a
total of three components per pixel ($2^{(4+4+4)}=2^{12}=12\text{bit}$, $2^{(6+6+6)}=2^{18}=18\text{bit}$).

The major let down by using 4-width is that it only works on PAL/NTSC/HighGFX and
that it's terribly slow. That's where the 1-width and 2-width modes come in. They
try to achieve the best possible quality with less pixels.

The 2-width uses a simple but very efficient theory about in which order the RGB components should be changed to get the best quality while only changing two of them per pixel.

The 1-width uses a similar theory (which I've invented myself) but ofcourse produces a worse quality, it also requires more computations per pixel and thus is slower, but since a PPC is very powerful, and the main bottleneck is the CHIP accesses, you might actually get a faster playback using 1-width.

I hope the text above explains why certain HAM widths only work with certain MPEG sizes, it's for example impossible to use 2-width on anything bigger than 640*512 since there are no available screenmodes, and if you use a VGA monitor, you can't display any screenmode larger than 640*512 (overscan not counted) which means that 2-width is only supported for MPEGs up to 320*256.

1.11 AmiDog's Movie Player - FAQ - HAM Quality

Back

Please read the text about HAM Width first if not already done.

Below you can read what the difference is when using High instead of Normal Quality.

2-width - The difference in quality is very tiny here. What AMP does is that it takes in account which RGB components that most urgently need to be changed in order to (mathematically) give the best result. On some MPEGs this can actually reduce the quality.

1-width - Here the difference in quality is quite big. When using high quality on this mode, AMP will use a predefined palette as well as using the normal "finetuning". For each pixel, AMP will calculate something called "color distance" to determine if it should use one of the 16/64 colors in the palette, or "finetuning". This ofcourse makes it even slower, but the quality is much better, and this mode works on every monitor, including VGA.

NOTE: 1-width high quality is the only mode which uses the palette, all others rely on "finetuning" for displaying the image.

1.12 AmiDog's Movie Player - FAQ - HAM S-Lores

Back

The Amiga actually supports something called Super Lores which aren't used very often. ↔

It's simply the same as a Lores screenmode but with half the height, i.e. PAL Lores is ↔

320*256, and PAL Super Lores is 320*128. I found out that Super Lores also work on Hires ↔

screens, i.e. PAL Hires Super Lores is 640*128.

By using these modes, AMP is able to display a MPEG up to 160*128 in fullscreen using ↔

2/4-width HAM since these modes requires a screen which is 2/4 times as wide as it's ↔

high. As a small bonus, these MPEGs will be played faster and in fullscreen.

The difference in speed between normal PAL Lores and PAL Super Lores, and which is most ↔

clearly visible in 4-width HAM, is due to less CHIP memory bandwidth being used by the ↔

custom chipset since the screen is smaller, this also gives an idea about how slow AGA ↔

is when using HAM8.

NOTE: Super Lores is only supported for PAL and 2/4-width HAM.

1.13 AmiDog's Movie Player - ShareWare

After your free trial period of 30 days, if you decide to keep using it, please send \$15, £10, 20DM, 100SEK or equal amount of any other currency to:

Mathias Roslund
Sveav. 2b, nb
S-702 14 Orebro
Sweden

Thanks!

NOTE: I need your name and e-mail address to be able to send you your keyfile!

1.14 AmiDog's Movie Player - Disclaimer

Remember! You use this piece of software at your own risk!

I can never be held responsible for any sort of damage caused to your software or hardware by the use of this product!

Bugreports and suggestions might be sent to one of my addresses.

1.15 AmiDog's Movie Player - Development

This product has been developed totally by me using EGCS/GCC.

Since I'm a student, I just don't have the time to spend several hours a day developing this product, especially when approaching Christmas and summer since I then will have a lot of schoolwork to finish. Therefore please don't write to me and complain about the slow development! Thanks!

AmiDog's Movie Player is developed using:

```
v0.00-0.31  Amiga1200HD -> 040/FPU/MMU'25, 603e'200, 2+32MB, AGA.  
v0.40-0.50  A1300Ti     -> 040/FPU/MMU'25, 603e'200, 2+128MB, AGA.  
v1.00-      A1300Ti     -> 040/FPU/MMU'25, 603e'200, 2+128MB, BVision.
```

1.16 AmiDog's Movie Player - Bugs

* The QT parser don't handle all QTs properly (I've found one which QT handles but both AMP and CyberQT fails with, not crashes, but gives bad audio/video).

There might be more bugs, so you use it at your own risk!

1.17 AmiDog's Movie Player - Contact

Bugreports, suggestions, comments or anything else you may want to contact me about can preferably be sent by e-mail to:

amidog@amidog.com

You may however also contact me by normal mail:

Mathias Roslund
Sveav. 2b, nb
S-702 14 Orebro
Sweden

1.18 AmiDog's Movie Player - Thanks!

I would like to thank the following persons:

- * Stefan Burström - For answering all my GCC PPC questions.
 - * Jesper Svennevid - For giving me the C-only C2P and helping me getting it to work. ←
 - * Mikael Kalms - For helping me getting the 4bit C2P to do only 4bit C2P.
 - * Steffen Haeuser - For always being very helpful.
- * Many thanks to all of you who have registered! Thanks!

1.19 AmiDog's Movie Player - Future

This is what I currently plan to add, it is NOT in priority order, and it might change without further notice!

- * Seeking in audio MPEGs with audio enabled.
- * Seeking in FLI/FLC/AVI/QT anims.
- * ANIM/CDXL... support.

1.20 AmiDog's Movie Player - History

v1.30 -991021
 -The CVID decoder didn't work, this has been fixed. I've also removed a bug ←
 from the
 CVID decoder which of some reason didn't do any harm.
 -The YUY2 decoder gave an "upside-down" image, fixed.
 -991027
 -Worked a bit on the audio routines and the new faulty timer, it's a bit ←
 better now, but
 on the other hand, the resync isn't done anymore.
 -991028
 -Finally got VLayer to work with AMP, the problem was that I wasn't using ←
 BPPC fix, and,
 of some reasons, cgxvideo fails if you use just "terminator 2" WarpUP ←
 setting. It's not
 very fast though, it's actually slower with some MPEGs than my custom ←
 routines. Ofcourse
 the scaling is a nice feature. VLayer is only supported for MPEG currently ←
 .
 -991101
 -Decided to make a big rewrite of some parts of AMP, the GUI is no more and ←
 a "MiniGUI" will
 be used instead (i.e. a popup window (RMB) giving access to the most ←
 usefull functions).
 -991107
 -The prefs has been totally changed, the CLI interface is new. After ←
 several hours of work,
 AMP now compiles nicely, I bet there are a lot of bugs to remove though.
 -After a few hours of bugtracing, AMP is actually starting to be usefull ←
 once again :) Anyway,
 there is a lot more to be rewritten/updated before making a new release.
 -991110
 -More rewrite, the code is getting a lot better structured, there are ←
 however very few of
 the original features which are working currently, but that WILL change!
 -Made several optimizations to the YUV->RGB conversion, and I just got over ←
 39fps
 while playing my test MPEG in a window on Workbench, and increase of over ←
 two fps.
 -991111
 -Moved refresh code into the MPEG video conversion routines (i.e. YUV->RGB/ ←
 HAM/Color/Gray)
 to gain some speed, and to make the general AmigaOS refresh routines much ←
 cleaner. The AGA

stuff is not tested, nor finished and do probably not work.
-For CGFX and MPEG the speedhack On/Off is now fully working.
-991114
-Faster overlay support. All modes not supported by speedhack will use CGFX ←
even when
speedhack is selected. Added gray support for window playback. Added gray ←
overlay support.
Some optimizations here and there in the MPEG display routines. Only AGA ←
support has to be
added and tested.
-Gray and color is now fully supported for AGA and I've started adding HAM ←
support as well.
The 2-width HAM is added and tested. It's however monday soon, so I better ←
get to bed now.
-991115
-Gray display on 15/16bit has been optimized and is now 15% faster than ←
before. The AGA
support (MPEG) is now finished and the HAM modes has been optimized as ←
well.
-I just made some new speed tests, and the optimized 2-width HAM routines ←
really are faster,
or what about 31.9/26.5 fps compared to 22.1/19.4fps, great isn't it! And ←
this is not all,
the 4bit gray increased from 40.2 to 49.7, and all the other AGA modes are ←
faster as well.
One nice thing is that with these new HAM routines, the 2-width HAM (6bit) ←
is faster than
the Color display was in v1.2x, and it gives a way better quality too!
-991117
-Cleaned up the entire MPEG video decoder source, this way, I removed a lot ←
of if's and made
it a little bit faster. I also made a small change to the IDCT routine. So ←
with my test MPEG
I now get 40.1 fps while playing it in a window on WB (16bit), a total ←
speedincrease by 3 fps
since the last release. The exe also got reduced by 30kb or so.
-I also removed a nasty bug, which really should have crashed AMP (I was ←
copying a string to a
memory area which hadn't been allocated), but it seemed to work anyway, ←
strange!
-MPEGs with strange widths wasn't displayed correctly when using window and ←
gray. I've also
optimized the gray overlay support a little.
-The MPEG decoder now only decodes the Y (luminance) values when gray is ←
selected, this
increased the 8bit gray (CGFX) speed from 57.6fps to 64.1fps, quite a ←
difference. The 4bit
gray (AGA) increased from 49.7fps to 54.8fps.
-991118
-Cleaned up all the AmigaOS code. I guess it's time to check if the audio ←
stuff is still
working, and to add stereo and dual mono support. I also need to make sure ←
that the FLI/FLC,
AVI and QT stuff is working properly (which I doubt). I mainly need to fix ←
the AGA stuff for
FLI/FLC/AVI/QT since I've made so many changes to the AGA code that it ←
wont work otherwise.

Then I can ofcourse clean up the MPEG audio code and add MP2/3 support.
-991119
-Hmm, I got Wipeout 2097 for my Amiga today, so it hasn't been much coding, ←
it's great by the way.
Anyway, it was ofcourse possible to remove more decoding when gray is ←
selected, so now the IDCT
isn't performed and the framerate increased from 64.1fps to 70.3fps using 8 ←
bit gray (CGFX).
There is still some things that's decoded to no use, I'll look deeper into ←
it tomorrow.
-991120
-Ok, now I've removed some more decoding of the UV components, and the ←
framerate for 8bit gray
(CGFX) is now up to 71.1fps. This means a total speedup from 57.6fps to ←
71.1fps or 23%.
There is still room for further speedups, but they would be quite small ←
and currently not
worth the effort. The best way would be to split the MPEG video decoder ←
into four different
versions, MPEG1 - color/gray and MPEG2 - color/gray, which could give ←
another 5-10% or so. This
will probably not happend since there are other things that can be ←
optimized and give a better
overall performace increase.
-Added a window to all screens opened by AMP, this way I can hide the ←
mousepointer and get
IDCMP messages, something that will be important in the future.
-Fixed the FLI/FLC/AVI/QT decoders to work with the new AmigaOS routines.
-Optimized and fixed the HAM routines currently used by AVI/QT. I've also ←
tried to use audio
with all types of movies and it appears to be working.
-The building of the QT indexes is now several hundred times faster due to ←
better fifo routines
using custom memorypools.
