

XMAME/XMESS DOCUMENTATION

Juan Antonio Martinez *jantonio@dit.upm.es* <mailto:jantonio@dit.upm.es> and Hans de Goede *hans@highrise.nl* <mailto:hans@highrise.nl> and Rene' Herrmann (4 doc/sgml) *rene.herrmann@berlin.de* <mailto:rene.herrmann@berlin.de> and Lawrence Gold (minor editing) *gold@aros.net* <mailto:gold@aros.net> and David Heremans (minor editing) v2.1.18, 1 August 2002

xmame/xmess documentation with an installation guide, the FAQ (frequently asked questions), hints, tips and tricks, the changes to the code, and much more!

Contents

1	Introduction	4
1.1	Welcome to XMAME	4
1.2	Keep everything up-to-date!	4
1.3	Pending Jobs - or TODO	5
2	Compiling, Setting Up and Starting Xmame/Xmess	5
2.1	Requirements	6
2.2	Follow these steps to compile xmame/xmess	6
2.3	If you have problems compiling, installing and starting xmame/xmess...	8
2.4	Compile Notes for Specific Operating Systems	8
2.4.1	Linux i386	8
2.4.2	Linux PowerPC	9
2.4.3	IRIX	10
2.4.4	HPUX	10
2.4.5	ULTRIX	10
2.4.6	SunOS & Solaris	11
2.4.7	Openstep	11
3	Xmame Frequently Asked Questions	12
3.1	What is mame?	12
3.2	What is mess?	12
3.3	What is xmame/xmess?	12
3.4	What are the differences between mame/mess and xmame/xmess?	12
3.5	What hardware and operating systems are supported?	13

3.6	My system is supported but xmame/xmess gives me errors and fails. Why?	15
3.7	What is the role of the xmamerc/xmessrc file?	15
3.8	What is the syntax of the xmamerc/xmessrc file.	15
3.9	Where do roms, images, samples, highscores, configfiles, etc go?	15
3.9.1	Roms/Samples	15
3.9.2	Read/write Diskette Images	16
3.9.3	Rom Cartridges/Read Only Diskette Images	16
3.9.4	Artwork	18
3.9.5	Xmame configuration	18
3.9.6	Game configuration	18
3.9.7	Game state	19
3.9.8	NVram	19
3.9.9	Mem cards	19
3.9.10	Screenshots	19
3.9.11	Highscores	19
3.9.12	hiscore.dat	20
3.9.13	cheat.dat	20
3.9.14	history.dat	20
3.9.15	Input logs	20
3.9.16	Xmame/xmess still can't find files/store highscores	20
3.10	How do I start xmess/xmame ?	20
3.11	What are the command line parameters for xmame/xmess?	21
3.12	What keys does xmame/xmess use?	21
3.13	Why doesn't a game have sound?	22
3.14	My sound seems lagged. / My sound Clicks. / What does -bufsize do?	22
3.15	Why doesn't a game work?	23
3.16	What are PROMs?	23
3.17	Why is xmame/xmess slow?	23
3.17.1	SPEEDUP # 1 - Inline compiling	24
3.17.2	SPEEDUP # 2 - Direct Graphics Access	24
3.17.3	SPEEDUP # 3 - NASM	25
3.17.4	SPEEDUP # 4 - ICC	25
3.18	I've got a dual processor machine. What can I do to take full advantage of this?	25
3.19	Why doesn't my joystick handle diagonal movement correctly on my Intel x86 system?	25

3.20	How do I enable XinputExtensions X11 based joystick in Xfree86?	25
3.21	How do I read a manual page?	25
3.22	Why do I get gz_open symbol not found while compiling?	25
3.23	Why do I get zlib.h not found?	26
3.24	When I run xmame.x11, nothing happens—no window pops up or anything. Help!	26
3.25	Can I run xmame fullscreen?	26
3.25.1	xmame.x11	26
3.25.2	xmame.sdl	27
3.26	xmame.SDL / fullscreen has a large border around the game.	27
3.27	I only use some of the emulation parts (Neo Geo / CPS-2 /...). Do I need the entire xmame?	27
3.28	Why does xmame freeze and lock up when I hit <ESC> to exit the game or P to pause the game?	28
3.29	Why does xmame report "can't resolve host name '/dev/dsp' when trying to use esound?	28
3.30	Where can I get the latest version of xmame/xmess?	28
3.31	I upgraded xmame and now some games do not work anymore.	28
3.32	Where can I send fixes, bug reports, suggestions, or money? :-).	29
3.33	What if my question is still not answered?	29
4	Display Subsystem-Specific Comments and Notes	29
4.1	FXmame, a glide driver for xmame	29
4.1.1	Requirements	29
4.1.2	Compiling and Installing	30
4.1.3	Running	30
4.1.4	Common Questions (FAQ)	30
4.1.5	Present Limitations/Bugs	30
4.1.6	Future Work	31
4.1.7	License	31
4.1.8	Version History	31
4.1.9	Author	31
4.2	GLmame, an OpenGL driver for xmame	31
4.2.1	Requirements	32
4.2.2	Compiling and Installing	32
4.2.3	Running	32
4.2.4	Cabinet Models	34
4.2.5	Common Questions (FAQ)	36

4.2.6	Present Limitations/Bugs	38
4.2.7	Future Work	38
4.2.8	License	38
4.2.9	Version History	38
4.2.10	Authors	39
5	Contact Information and Credits	39
5.1	Mail	39
5.2	Bugzilla	41
5.3	Credits	41
6	Legal Issues and Copyrights Concerning MAME/MESS	43
7	Some Links	43
7.1	Unix sources and docs	43
7.2	Unix binaries	43
7.3	Xmame/xmess download mirrors	44
7.4	(X)Mame front-ends	44
7.5	Some other links	45
8	ChangeLogs	45
8.1	The MAME/MESS ChangeLogs	45
8.2	The Unix ChangeLog	45

1 Introduction

1.1 Welcome to XMAME

This is the the *nix port of the almost legendary mame. Mame is an arcade machine emulator, started in 1997 by Nicola Salmoria. It started out as a series of emulators for individual games. This series of emulators was combined into a single multi-game emulator. This is the current form of mame; no longer a one-man show, there are over 100 contributors to the project.

1.2 Keep everything up-to-date!

The most recent copy of this guide, the xmame-sources and many more can be found on the xmame project homepage at <http://x.mame.net/> <<http://x.mame.net/>>

1.3 Pending Jobs - or TODO

If you think you can help with the todo list or you can make a contribution that you feel is interesting, please send mail to the xname/xmess coordinator. See 5.1 (Mail) for contact information.

- Optimise mouse warping.
- Remove `osd_input_init_post`, add display ready flag.
- Updating of fps is off (dirty marking problem) with vector games.
- Update `porting.txt`.
- Convert all remaining sound drivers to plugins.
- Rewrite display/input code to be more modular, oo-ish, and usable as a separate library.
- Split `-[no]scanline` option into emulated and real scanline options.
- Add an option to specify which resolution to use, add clipping and panning if it doesn't fit and use it anyway.
- Use Pentium TSC for `uclock` where available.
- Network:
 - Fix since the dos team broke it, find out if the dosnet team is still alive.
 - Add mouse/joystick support to network play
 - Try converting network support to tcp, currently we have no error correction.
 - Make network code more robust, version check, input settings check, etc.
- Check `x11-joystick`. Rumor has it it's broken, but then again there's only one bug-report, so it seems no one is using it.
- fm soundhack for linux-i386.
- Make which messages are emitted and when configurable (log-type's and log-level), make `fileio` emit messages about which files it tries to open, to help debug rom-loading problems, and walk through all messages `xname` emits giving them all a standard format like:

```
OSD:      Info:    bla
X11-window: Warning: bla
Svgalib:  Error:   bla
```
- Add scancode 128-255 for non-PC keys like F13 – F24, more keypad keys, etc., update all display targets to use this.
- Also see the text `todo` file, kept around to quickly enter stuff while hacking: `todo <todo>`.

2 Compiling, Setting Up and Starting Xname/Xmess

Note: This section is not a Unix tutorial. You need to know some basic Unix commands to compile and install `xname/xmess`.

2.1 Requirements

- GNU make is MANDATORY in ALL Systems, even if you use a different compiler.
- On all platforms zlib is now needed since the core uses it. If you don't have zlib, a stripped version is included with xname. See: 3.22 (FAQ: Why do I get gz_open symbol not while compiling?) and 3.23 (FAQ: Why do I get zlib.h not found?)
- You are encouraged to use a GNU development environment: gcc, gmake, gnu- ar. Lots of problems are found due to the very permissive syntax in gcc, so if you use a strictly ANSI C compiler, you'll get lots of warnings, and if the version of xname/xmess is not fully debugged, compilation errors.
- Xname compilation breaks when using egcs-1.1.x or gcc-2.95, use `-fno-strict-aliasing` to fix it. gcc-2.95.1 does work with `-fstrict-aliasing`, and gives a nice speed increase.
- Xname compilation also breaks when using Red Hat 7's latest (as of 16 January 2001) update for gcc. If you experience problems, please either downgrade your compiler to the one included on the Red Hat 7 CD-ROM, or try replacing "gcc" in the Makefile with "kgcc", which will invoke egcs-1.1.2. (But please see above about the `-fstrict-aliasing` flag.)

2.2 Follow these steps to compile xname/xmess

1. See 2.4 (Compile Notes for Specific Operating Systems) for platform specific compiling information.
2. See 4 (Display subsystem specific comments and notes) for special display methods information.
3. Download the xname/xmess source files archive.
4. Move the xname/xmess source files archive to a suitable work directory. When you extract the archive, sub directories will be created under the work directory.
5. Rename the source files archive so its extension is .gz, so that gzip doesn't complain. For example, change `xname-0.34b1.1.tgz` to `xname-0.34b1.1.tar.gz`
6. Use `gzip -d` to uncompress the archive. For example, type:

```
gzip -d xname-0.34b1.1.tar.gz
```

7. Extract the files with tar. For example, type:

```
tar xvf xname-0.34b1.1.tar
```

8. After extracting the archive, change to the subdirectory created by the extraction process. For example, after extracting the `xname-0.34b1.1.tar` archive, a directory named `xname-0.34b1.1` is created. Change to it. This directory is the top of the source tree.
9. Copy `makefile.unix` to `Makefile`, the distributed `Makefile` is used for development and is not a good starting point!
10. Edit the `Makefile` and change the settings according to your system. The default `makefile.unix` is preconfigured to run under X11/Linux. Be careful editing this file.

- If you intend to compile in X11 joystick support, type the following to see the available Xdevice list:

```
make xlistdev; ./xlistdev
```
- Choose the desired device and re-edit Makefile, setting the option X11_JOYNAME to the correct device.

11. Compile xname/xmess by typing:

```
make
```

12. If the compile fails, report the problem to the author.

13. ***** YOU MAY NEED TO BE root TO PERFORM THIS STEP ***** To install xname/xmess, type:

```
make install
```

14. If you want you can now build xname/xmess for another target/display method, otherwise skip to the next step.

- (a) If you want to also build for another target repeat repeat the "make" and "make install" steps adding "TARGET=<target>" to the make command for example:

```
make TARGET=mess
make install
```

- (b) If you want to also build for another display method repeat the "make" and "make install" steps adding "DISPLAY_METHOD=<method>" to the make command for example:

```
make DISPLAY_METHOD=svgalib
make install
```

- (c) You can also combine the 2 above for example:

```
make TARGET=mess DISPLAY_METHOD=svgalib
make install
```

15. In your home directory, create a subdirectory called .xname.

16. From the top of the source tree, copy the file doc/xnamerc.dist to the `/${HOME}/.xname/` directory. (For xmess do this for xmessrc)

17. If desired, copy xnamerc.dist to the global xnamerc location as "xnamerc" (for xmess, copy xnamerc.dist to xmessrc). The global xnamerc location is defined in the Makefile.

18. Edit `/${HOME}/.xname/xnamerc` to match your preferences. (For xmess, edit `/${HOME}/.xmess/xmessrc`)

19. Be sure that the DISPLAY and PATH environment variables are set correctly.

20. Make sure your ROM images are installed.

21. Start the X-Window System. (Unless you compiled for a different display method in which case you skip this step.)

22. Start xname/xmess. For example:

```
/usr/games/xname.x11 dkong
```

2.3 If you have problems compiling, installing and starting xname/xmess...

- Read 3 (Xname frequently asked questions)
- Read (the recent parts of) 8 (Changelogs). They might contain some hints about recent changes which aren't fully documented yet.
- Check 2.4 (Compile Notes for Specific Operating Systems) for operating system-specific notes. You may need to do something special for your operating system.
- Check 4 (Display subsystem specific comments and notes) for display subsystem-specific notes. You may need to do something special for your display system.
- If you get an error about `gz_open` and a bunch of other functions not found during linking, read: 3.22 (FAQ: Why do I get `gz_open` symbol not while compiling?) and 3.23 (FAQ: Why do I get `zlib.h` not found?).
- Be sure that the `rompath` in your `xnamerc/xmessrc` is correct; see 3.9 (FAQ: Where do `roms.images.etc` go?).
- Check `xnamerc/xmessrc`: Is it configured correctly?
- Are you getting messages that you are missing files? Maybe you are missing PROMs; see 3.15 (FAQ: Why doesn't a game work?) and 3.16 (FAQ: What are PROMs?)
- Is the program in your `PATH`?
- The `svgalib/xf86_dga` build of `xname/xmess` requires `setuid root`. Are your permissions correct?
- Try asking the friendly folks at the newsgroup `alt.game.name` to see if anyone else has experienced the same problem as you.
- Try mailing to the `xname` mailinglist, see 5.1 (Mail) for more information.

If all else fails, send email to the current maintainer of the `xname/xmess` project. See 5.1 (Mail) for contact information.

2.4 Compile Notes for Specific Operating Systems

2.4.1 Linux i386

Linux i386 is the platform I (Hans) use, so `xname` should compile without any hassles on Linux i386. (Note: The same is true for Lawrence Gold.)

Some bugs have been found and worked around when using `gcc27x`, please use `egcs` if available. If more `gcc` bugs popup, `egcs` will be made mandatory, but I hope that won't be necessary.

It has been reported that some SoundBlaster AWE32 and VIA-82cxxx (onboard) cards do not correctly manage `get_freespace()`. If so, you'll get some `audio_ioctl()` calls failed messages in console, and if everything goes right a message that `xname/xmess` is falling back to timer based audio. If this doesn't happen and sound isn't right, try forcing timer-based audio by starting `xname/xmess` with `-timer`.

SB64/128PCI cards (Ensoniq 137x-based cards) also do not correctly manage `get_freespace()`. Unfortunately, this doesn't result in failed `ioctl`'s; instead, they just ignore the calls. `Xname` tries to autodetect this

and to fall back to timer-based audio. If this doesn't happen and sound isn't right, try forcing timer based audio by starting xname/xmess with `-timer`.

Some have reported success using ALSA with OSS emulation (see <http://www.alsa-project.org/> <<http://www.alsa-project.org/>>). If you experience the above problems, this would certainly be worth a try.

Also, I've tested sound using the PC-Speaker driver module. It works!! But it needs `-timer`. Again, it should do this automatically, but you know what to do if it doesn't.

Distribution-Specific Comments:

- S.u.S.E. Linux: Besides the gcc compiler and libc libraries, which are both from the "d" series, other packages need to be installed or the build will fail. If compiling for svgalib, the svgalib package from series "d" installed. If compiling for the X-Window System, you must have x-devel package from series "x" installed. These packages can be installed using the "yast" utility included in the S.u.S.E. distribution.
- Red Hat Linux is the operating system used by the author. xname/xmess should build correctly under Red Hat Linux. :-)

2.4.2 Linux PowerPC

Linux PowerPC should work from now on. Here's an email from Andreas Varga sid@skater.htu.tuwien.ac.at <<mailto:sid@skater.htu.tuwien.ac.at>>, the Linux/PowerPC maintainer, describing the necessary steps:

Required changes to compile xname 0.34b3.1 on Linux/PowerPC with SVGALIB

Makefile:

- enable CFLAGS for linux/ppc
- enable DISPLAY_METHOD = svgalib
- enable ARCH = linux
- remove `-lvga` from LIBS.svgalib in `src/unix/unix.mak` (only `-lvga` is required)

Required changes to compile xname 0.34b3.1 on Linux/PowerPC with X11

Makefile:

- enable CFLAGS for linux/ppc
- enable DISPLAY_METHOD = x11
- enable ARCH = linux

Another thing... sound is not working, because sound drivers support is currently under development in Linux/PPC kernels...there is an old driver that is a hacked Soundblaster driver, and newer 2.1.1xx kernels have a sound drivers based on Amiga DMA sound drivers... Once this issues are settled there should be a full OSS-compatible driver... sound should work then...

I can't speak about Joystick support...

The original Linux/PPC target changes in the Makefile were done by me, but there are some other changes required to get it to compile... Starting from October I will have some more time to work on this, and I'll keep you updated...

Cheers, Andreas

2.4.3 IRIX

Use the generic unix arch for those machines that have no sound support installed (xname/xmess uses the dmedia package or the newer AL package on IRIX systems).

Silicon Graphics provides several native compilers. xname/xmess has been compiled and tested with the normal cc. There were some problems with this when using -n32 for new style 32bit code. It does work when using -o32 for oldstyle 32 bit code. It should now also work with -n32 which is much faster.

The use of DCC is not recommended (anymore) since it's a C++ compiler and thus won't even compile xname/xmess out of the box due to uncasted mallocs. Changing all these mallocs would be crazy, esp. since they are not only in the osd part, but also in the core which is not under our control. Besides that, name is normal C, so please compile it with a normal C compiler.

Some problems have been reported using the GNU linker. It's recommended that you use the native one.

2.4.4 HPUX

Use the generic unix arch. In HP-UX, gcc and gnu-make are MANDATORY.

2.4.5 ULTRIX

Use the generic unix arch.

Notes received from Danny *dboxhoor@iso.vilspa.esa.es* <mailto:dboxhoor@iso.vilspa.esa.es>

VERY IMPORTANT!!!

The xname/xmess CPU emulators access memory through pointers which are sometimes cast to illegal addresses - as far as the MIPS CPU is concerned. The Ultrix operating system is aware of that and fixes this "unaligned data access".

Depending on your system setup, lots of warnings are generated for certain games. Fortunately these messages can be disabled by executing the command "uac p 0" in the same shell where xname/xmess will be running.

UPDATE: The new -DALIGN_INTS switch in the makefile fixes these unaligned accesses in the source which is probably a lot faster than hot fixing them in the kernel segfault handler.

Use gnu make.

Compile with the 'c89' compiler. Though 'gcc' has no problems with compiling and optimizing xname/xmess, the resulting 'c89' executable is faster on an empty machine (read: xname can use all cpu & memory without competing with other high priority processes), so you can set 'frameskip' to 1.

When compiling with 'c89 -O2' don't forget to also use '-Olimit 1000'. If 1000 is not enough, the compiler will tell you to increase it.

2.4.6 SunOS & Solaris

There are 2 relevant architecture options available in the makefile:

```
ARCH = generic : use this for old SunOs running on Sparcs,  
                but a Sparc running Solaris will usually  
                handle this option too.  
ARCH = solaris : this is the one you should use for Sparcs  
                running Solaris.
```

Currently sound is not supported under sunos. The solaris arch does support audio.

If you are using Sun's OpenWindows implementation of X11, then make sure that the X11INC and X11LIB variables in the X11 Options section of the makefile are set properly. (There are already entries for the standard OpenWin directory there that you can uncomment.)

Of course, if you are compiling for X11 then you need to set these variables properly, no matter what implementation of X11 you are using! :)

If you are using Sun's SunPro cc compiler, then the following optimisation flags have been reported to be good:

```
OPTFLAGS      = -fast -x04 -native
```

Finally, there has been a case of the xname compilation processes breaking over the "ar" program. Most Solaris installations will include a set of utilities that conform to the XPG4 standard, usually in the "/usr/xpg4/bin" directory. The XPG4 version of "ar" should be avoided like the plague. Ensure that the "/usr/ccs/bin" appears in your path before the "/usr/xpg4/bin" directory and everything should go smoothly.

PLEASE NOTE : There is currently not much support for SunOS/Solaris. If you have any trouble, try to fix it yourself. If you succeed, send a message to the mailing list detailing what you did. If you don't succeed, send a message anyway and someone may be able to help you. Either way, this is the only way that problems with xname on SunOS/Solaris will get fixed.

2.4.7 Openstep

You must use the OpenStep gnumake, not make. You must also use the NeXT/Apple cc rather than gcc. Also use only the optimisation options as recommended for OpenStep in the makefile, others will cause the resulting executable to crash. The OpenStep assembler on i386 architectures is incompatible with the mame assembler CPU cores and thus you must use the C versions (Use i386-noasm instead of i386 when compiling for i386).

The supplied compiler is based on gcc 2.7.2.1 which will not compile src/profiler.c without crashing. To get around this you must turn off + optimisation when this crash occurs, re-make to compile this file only, then stop the make, re-enable optimisation, and continue.

On an OpenStep keyboard the functions keys do not all operate as expected. Using the command key in conjunction with a number key generates the equivalent function key within xname, with 0, - and = + representing F10, F11 and F12 respectively. Also note that the ALT + keys cannot be mapped. As most games tend to use this as the default fire button it is extremely useful to remap this globally the first time you run xname.

3 Xname Frequently Asked Questions

3.1 What is mame?

Mame is an arcade machine emulator. Started in 1997 by Nicola Salmoria, mame started out as a series of emulators for individual games. This series of emulators was combined into a single multi-game emulator. This is the current form of mame; no longer a one-man show, there are over 100 contributors to the project.

Mame was created by Nicola Salmoria.

Also see *the official mame readme file (readme.mame)* <readme.mame>

3.2 What is mess?

Mess is just like mame—a virtual machine emulator, only it doesn't emulate arcade machines but rather computers and consoles.

Also see *the official mess readme file (readme.mess)* <readme.mess>

3.3 What is xname/xmess?

Xname/xmess are the Unix/X11 ports of the mame and mess projects. It makes mame/mess available on *ix machines using the X11R6 X-Window System (and SVGAlib/ggi/XF86-DGA/OpenGL/SDL too).

Mame was originally ported by Allard Van Der Bas, Dick the Ridder, Juan Antonio Martinez, and Hans de Goede.

Xname/xmess is currently maintained by Lawrence Gold.

3.4 What are the differences between mame/mess and xname/xmess?

None significant.

Xname/xmess is based on the mame/mess source code. Due to technical reasons beyond the scope of this document, the mame/mess source may not compile under Unix. That's why the xname/xmess project exists. Each time mame/mess is updated, the code is tested (and patched if needed) under Unix. This way xname/xmess releases are always the same as their mame counterparts.

There are no plans for the independent development of xname/xmess.

3.5 What hardware and operating systems are supported?

Xmame runs on both little-endian and big-endian CPUs, using aligned integer access if needed and is compatible with 64 bit CPUs. The current list of supported CPUs in the makefile is:

- i386 + gnu-asm
- i386 no asm
- alpha
- ia64 (merced)
- m68k
- generic risc (PowerPC, Sparc, HPPA, IBM)
- generic risc, lsb-first (RISC (Ultrix machines))
- mips (generic risc + SGI compiler bug workarounds)

Adding support for another cpu (if necessary) should be trivial.

The following Unix platforms are specifically supported:

- Linux
- FreeBSD
- NetBSD
- Solaris (SunOS)
- OpenStep (noasm only on i386)
- IRIX with sound using the dmedia package or the new al package (sound currently broken)
- AIX (sound currently broken)
- generic unix, no sound

The generic unix entry should work for any reasonable standard Unix variant, but lacks sound since there is no single Unix sound standard.

On all platforms, zlib is now needed since the core uses it. If you don't have zlib, use the included one by uncommenting the appropriate line in Makefile.

The following display methods are supported. X11 is the only one which works on all platforms, except for OpenStep which only works with Openstep bitmaps:

- X11 (X-Window System)
- svgalib
- GGI (Generic Graphics Interface)

- OpenGL using X11 for input
- glide using X11 for input (3DFX)
- glide using svgalib for input (3DFX)
- OpenStep bitmaps
- SDL (Simple Direct Medialayer)

The following is a list of arch/os/display combinations, with maintainers, for arch/os/display combinations which are actively supported. Please mail the maintainer of the arch/os/display combination for compilation and other arch/os/display specific problems. Mail Lawrence gold@aros.net <mailto:gold@aros.net> for generic (e.g., X11) problems.

- i386/linux/x11 (+DGA) -> Lawrence <gold@aros.net> <mailto:gold@aros.net>
- i386/linux/svgalib -> Lawrence <gold@aros.net> <mailto:gold@aros.net>
- i386/linux/xgl -> Sven Goethel <sgoethel@jausoft.com> <mailto:sgoethel@jausoft.com>
- i386/linux/xfx -> Lawrence <gold@aros.net> <mailto:gold@aros.net>
- i386/linux/svgafx -> Lawrence <gold@aros.net> <mailto:gold@aros.net>
- i386/linux/ggi -> Christian <cpg@aladdin.de> <mailto:cpg@aladdin.de>
- alpha/linux/x11 -> Christian <cpg@aladdin.de> <mailto:cpg@aladdin.de>
- alpha/digital unix/x11 * -> Leandro <gold@aros.net> <mailto:gold@aros.net>
- powerpc/linux/x11 -> Andreas <sid@skater.htu.tuwien.ac.at> <mailto:sid@skater.htu.tuwien.ac.at>
- powerpc/linux/svgalib -> Andreas <sid@skater.htu.tuwien.ac.at> <mailto:sid@skater.htu.tuwien.ac.at>
- i386/freebsd/x11 -> Jacob A. Hart <c9710216@studentmail.newcastle.edu.au> <mailto:c9710216@studentmail.newcastle.edu.au>
- i386/netbsd/x11 -> Dave <dave@dtsp.co.nz> <mailto:dave@dtsp.co.nz>
- alpha/netbsd/x11 -> Christian <cpg@aladdin.de> <mailto:cpg@aladdin.de>
- Sparc/Solaris/x11 -> Lawrence <gold@aros.net> <mailto:gold@aros.net>
- i386/Unixware7/x11 * -> Richard <xmame@frogface.ddns.org> <mailto:xmame@frogface.ddns.org>
- i386/openstep -> Pete French <pete@toybox.twisted.org.uk> <mailto:pete@toybox.twisted.org.uk>
- Sparc/SunOs -> Saga <jolletx@cybercable.fr> <mailto:jolletx@cybercable.fr>
- mips/irix/x11 -> Dingus Family <ddingus@uswest.net> <mailto:ddingus@uswest.net>

* These are supported using the generic unix entry of the makefile, and thus lack sound.

We are always looking for people willing to test and answer questions about xmame on a new arch/os/display. This takes very little work and could help other people tremendously. If you're interested, mail Lawrence gold@aros.net <mailto:gold@aros.net>.

3.6 My system is supported but xmame/xmess gives me errors and fails. Why?

Xmame/xmess should run on 8bpp Pseudo Color displays and on 16, 24 and 32bpp True Color displays. If your X-Server does not support any of these display types, xmame/xmess will not run.

You can verify your X-Server support by starting your X-Server and running the xdpinfo program. See the xdpinfo manual page for more information.

3.7 What is the role of the xmamerc/xmessrc file?

A normal xmame/xmess invocation looks like:

```
xmame pacman -sound -nojoy -frameskip 2 -heightscale 3 \  
-widthscale 2
```

Most of the time, you will want to use the same configuration for each game. xmame/xmess will read this file for default values for the parameters that you can set. This way the only parameters you need to express on the command line are the ones you want to override.

Another reason for xmamerc/xmessrc is keyboard remapping. Some Unix keyboards are very different than their PC counterparts. To compensate for the changes, xmame/xmess allows you to remap keys that are not available on Unix keyboards to keys that are available.

3.8 What is the syntax of the xmamerc/xmessrc file.

The syntax is quite simple:

```
"keyword value"
```

See *xmamerc.dist* <xmamerc.dist> for most keywords. Almost all commandline options can be used as keywords (without the -), use value 0 or 1 for commandline options which have a negating (-no[option]) counterpart. As usual lines beginning with "#" are comments.

You can also generate a configfile with all the current settings to use as a template by typing:

```
xmame -showconfig > /.xmame/xmamerc
```

3.9 Where do roms, images, samples, highscores, configfiles, etc go?

3.9.1 Roms/Samples

Xmame/xmess uses a ":" separated rompath to find out which dirs to search for roms/samples.

The default rompath is XMAMEROOT as defined in Makefile during compile.

This is overwritten by any rompath in xmamerc/xmessrc, which can be overwritten again with the -rompath commandline option.

Xname/xmess searches every dir in this path for roms/samples in the following way:

```
<dir>/gamename.zip           (containing filename.ext)
<dir>/gamename/filename.ext
<dir>/gamename/filename.ext.gz (containing filename.ext)
<dir>/gamename/filename.zip   (containing filename.ext)
```

So let's say that we are searching for rom0001.1, which is part of pacman. Then xname/xmess would search every dir in your rompath for

```
<dir>/pacman.zip             (containing rom0001.1)
<dir>/pacman/rom0001.1
<dir>/pacman/rom0001.1.gz    (containing rom0001.1)
<dir>/pacman/rom0001.zip     (containing rom0001.1)
```

Note: If your neogeo games can't find neogeo.rom, put it in a dir called neogeo somewhere in your rompath, or put it in neogeo.zip somewhere in your rompath.

3.9.2 Read/write Diskette Images

This section only applies to xmess emulations which support read/write diskette images and thus try to open the image read/write.

Xmess first searches for read/write images with the name specified on the commandline:

```
<path>/image.ext
```

where path is the path specified with the image on the commandline. This way, absolute and relative filenames can be used directly on the commandline for an image which is not in the rompath.

Then xmess searches every dir in the rompath for roms/samples in the following way:

```
<dir>/image.ext
```

Note that (g)zipped images are not supported for r/w images.

3.9.3 Rom Cartridges/Read Only Diskette Images

Xmess begins by searching for cartdriges/disks from the current dir in the following way:

```
<path>/image.ext
<path>/image.ext.gz       (containing image.ext)
<path>/image.zip          (containing image.ext)
```


where path is the path specified with the image on the commandline. This way, absolute and relative filenames can be used directly on the commandline for an image which is not in the rompath.

Then xmess searches every dir in the rompath for carts/images in the following way:

```
<dir>/image.ext
<dir>/image.ext.gz          (containing image.ext)
<dir>/image.zip             (containing image.ext)
```

Finally, xmess searches the rompath as described above using the systemtype for gamename.

So let's say we want to play mario.nes on the nes emulation of xmess. We would then type: "xmss nes mario.nes". Xmess then begins by searching the current dir for

```
mario.nes
mario.nes.gz              (containing mario.nes)
mario.zip                 (containing mario.nes)
```

After that, xmess searches each dir in your rompath for

```
<dir>/mario.nes
<dir>/mario.nes.gz       (containing mario.nes)
<dir>/mario.zip          (containing mario.nes)
```

And finally, xmess searches each dir in your rompath for

```
<dir>/nes.zip            (containing mario.nes)
<dir>/nes/mario.nes
<dir>/nes/mario.nes.gz  (containing mario.nes)
<dir>/nes/mario.zip     (containing mario.nes)
```

Since xmess first searches for cartridges/disks from the current dir you can also specify cartridges/disks not in your rompath. For example:

```
"xmss nes ../mario.nes"
```

Xmess then searches for

```
../mario.nes
../mario.nes.gz        (containing mario.nes)
../mario.zip           (containing mario.nes)
```

and will also search the rompath as described above with ../mario.nes as the image name but this is not that useful.

3.9.4 Artwork

Xmame/xmess searches every dir in the rompath for artwork in the following way:

```
<dir>/artwork.png
<dir>/artwork.png.gz      (containing artwork.png)
<dir>/artwork.zip         (containing artwork.png)
```

Then xmame/xmess searches the rompath as in roms/samples.

So let's say we want to play foo which uses bar.png. Xmame/xmess then begins searching each dir in your rompath for

```
<dir>/bar.png
<dir>/bar.png.gz          (containing bar.png)
<dir>/bar.zip             (containing bar.png)
```

And then xmame/xmess searches each dir in your rompath for

```
<dir>/foo.zip             (containing bar.png)
<dir>/foo/bar.png
<dir>/foo/bar.png.gz     (containing bar.png)
<dir>/foo/bar.zip        (containing bar.png)
```

3.9.5 Xname configuration

Xmame looks for the following configuration files and parses each one found. Files parsed later can overwrite options in previously parsed files:

```
XMAMEROOT/xmamercc
XMAMEROOT/xname-DISPLAY_METHODrc
${HOME}/.xname/xmamercc
${HOME}/.xname/xname-DISPLAY_METHODrc
${HOME}/.xname/rc/<gamename>rc
```

XMAMEROOT is defined during compile time, see Makefile.

DISPLAY_METHOD is the DISPLAY_METHOD for which xmame was compiled, this can be one of: x11, svgalib, ggi, xgl, xfx, svgafox, openstep, SDL.

For xmess of course replace xmame by xmess. ;)

3.9.6 Game configuration

Game configuration files are kept on a per-user base in:

```
${HOME}/.xname/cfg/<game>.cfg resp
${HOME}/.xmess/cfg/<game>.cfg
```

3.9.7 Game state

Game state files are kept on a per-user base in:

```
`${HOME}/.xname/sta/<game>.sta resp  
`${HOME}/.xmess/sta/<game>.sta
```

3.9.8 NVram

NVram files are kept on a per-user base in:

```
`${HOME}/.xname/nvram/<game>.nv resp  
`${HOME}/.xmess/nvram/<game>.nv
```

3.9.9 Mem cards

Mem cards files are kept on a per-user base in:

```
`${HOME}/.xname/mem/<filename>.mem resp  
`${HOME}/.xmess/mem/<filename>.mem
```

Where `<filename>` is given by the game-driver.

3.9.10 Screenshots

Screenshots, which can be made by pressing F12, are saved as png files in the directory from where xname is started (cwd).

This can be overridden by the `screenshotdir` entry in `xmamerc/xmessrc`, which can again be overridden by the `-screenshotdir` cmdline-option.

They are saved as:

```
<screenshotdir>/<game>xxxx.png  
Where xxxx is a number.
```

3.9.11 Highscores

Highscores are stored in `XMAMEROOT` as defined in Makefile when building. This can be overridden by the `spooldir` entry in `xmamerc/xmessrc`, which can again be overridden by the `-spooldir` cmdline-option.

So only give a

```
drwxrwsr-x root games <spooldir>
```

permission and `setgid` `xname/xmess` to "games" to get it to work. Note this doesn't work with `suid root` versions (svgalib & XF86-DGA), suggestions welcome.

For now use ``${HOME}/.xname` or ``${HOME}/.xmess` as the `spooldir` in `xmamerc/xmessrc` for `suid root` versions.

Note: For highscores to work, `xname` must also be able to find `hiscore.dat`.

3.9.12 hiscore.dat

If you want to use highscores mame needs to load hiscore.dat to get highscore info.

The default is XMAMEROOT/hiscore.dat . XMAMEROOT is set during compile-time; see the Makefile. This can be overridden by the hiscorefile entry in xmamerc if present, which can again be overridden by the cmdline-option `-hiscorefile`.

hiscore.dat is available from the mame homepage: <http://www.mame.net> <<http://www.mame.net>>

3.9.13 cheat.dat

If you want to use cheats mame needs to load cheat.dat to get cheat info.

The default is XMAMEROOT/cheat.dat . XMAMEROOT is set during compile-time; see the Makefile. This can be overridden by the cheatfile entry in xmamerc if present, which can again be overridden by the cmdline-option `-cheatfile`.

cheat.dat is available from the mame homepage: <http://www.mame.net> <<http://www.mame.net>>

3.9.14 history.dat

If you want to display some nice background info on games, mame needs to load history.dat.

The default is XMAMEROOT/history.dat . XMAMEROOT is set during compile-time, see the Makefile. This can be overridden by the historyfile entry in xmamerc if present, which can again be overridden by the cmdline-option `-historyfile`.

history.dat is available from the mame homepage: <http://www.mame.net> <<http://www.mame.net>>

3.9.15 Input logs

Input logs are saved/read under the name specified after the `-record / -playback` option; this name is relative to the current path.

3.9.16 Xname/xmess still can't find files/store highscores

Type "`xname/xmess -showconfig | more`" and check that the rompath and spooldir shown are correct and readable/writable. Otherwise recheck xmamerc/xmessrc. Remember that xmess also searches the current dir.

3.10 How do I start xmess/xname ?

To start xname, type

```
xname [options] <gamename> [more options]
```

For xmess, type

```
xmess [options] <systemname> [more options] \
<disk/romimage(s)> [more options]
```

All options are optional. ;)

Xname/xmess supports most dos command line options (see `readme.name` / `readme.mess`) as well as a few of its own: Type `xname -help` | `more` for a complete list.

With `xmess`, some systems don't require a `disk/romimage`, but most do.

3.11 What are the command line parameters for `xname/xmess`?

To see a list of available parameters, type:

```
/usr/games/xname -help | more
```

Xname/xmess supports most of the command line parameters as the DOS version as well as some of its own. See the file `readme.name` / `readme.mess` for more information about the command line parameters.

3.12 What keys does `xname/xmess` use?

Xname/xmess uses the same key-bindings as the DOS version of `mame`: The following keys work in all emulators:

' or	Enter On screen display to change volume, brightness etc.
Tab	Enter configuration menu. Press Tab or Esc to get back to the emulation.
P	Pause
F3	Reset
F4	Show the game graphics. Use cursor keys to change set/color, F4 or Esc to return to the emulation.
F8	Reduce frame skip
F9	Increase frameskip
F10	Toggle speed throttling
F10 + shift	Toggle idle sleeping
F11	Toggle speed display
F11 + shift	Toggle profiler display
F12	Save a screen snapshot
ESC	Exit emulator

Xname has 2 different auto frameskipping algorithms. These can be selected with the `-frameskipper` option and with key combos:

Key combo:	Gives you the:
<code>left-ctrl + insert</code>	standard dos frameskipper
<code>left-ctrl + home</code>	William Barath's frameskipper

Xname's display scaling can be controlled on the fly with the following key combos:

Key combo:	Result:
<code>left-shift + insert</code>	Increase widthscale
<code>left-shift + delete</code>	Decrease widthscale
<code>left-shift + home</code>	Increase heightscale
<code>left-shift + end</code>	Decrease heightscale
<code>left-shift + pageup</code>	Increase scale
<code>left-shift + pagedown</code>	Decrease scale

Under X11, xname can switch to and from different modes during runtime:

Key combo:	Warps you to:
<code>left-alt + insert</code>	Normal window
<code>left-alt + home</code>	DGA fullscreen mode

Under SDL, xname can switch to and from different modes during runtime:

Key combo:	Result:
<code>left-alt + enter</code>	Toggles between windowed and fullscreen modes.

Under X11 in a window, you can also grab the mouse. To be able to play with the mouse, press `left-alt + page-down` to grab the mouse, and press it again to release it.

3.13 Why doesn't a game have sound?

- First check that sound is working in other applications.
- Check that sound is supported for the game that has no sound; not all games have sound.
- For xname, take a look at `gamelist.mame`: Is the sound for the game emulated?
- For xmess take a look at `readme.mess`: Is the sound for the system emulated?
- Check the operating system support chart in the earlier part of this file: Does xname/xmess support sound on your system?

3.14 My sound seems lagged. / My sound Clicks. / What does `-bufsize` do?

With the `-bufsize` commandline option or the `bufsize` configfile keyword, you can specify the number of frames of audio to buffer. The default is 2 frames.

This value should be ok for everyone, but if you still have problems, read on.

If the buffersize is too small, this causes the buffer to get empty between frames which in turn causes clicks/pauses in the sound. If this is the case, increasing the `-bufsize` value to, for example, 2.5 should help.

If, however, you make the buffersize too large, then so much audio becomes buffered, that the sound could become lagged. E.g., you would hear the coin drop sound 0.5 seconds after pressing the insert coin key. In this case, lower the bufsize to, for example, 1.5.

Please let me know if you need to change this setting, since I'm planning on removing it in a future release.

3.15 Why doesn't a game work?

- Do you have the right roms or did you get checksum failures?
- Xname version 0.34 and higher require PROM files that were not included in ROM images released before version 0.34. See the question on PROMs below for more information.
- For xname, check the file gamelist.mame to see if the game is playable. Just because a game is included doesn't mean it is working yet.
- For xmess, take a look at readme.mess. Maybe the system isn't emulated all that well yet.
- Some games require sound support. If you don't have sound support under your Unix installation or sound isn't supported by xname/xmess for you, try the `-fakesound` command line parameter.

3.16 What are PROMs?

PROMs contain game-specific information that used to be built in to mame in versions prior to 0.34. This information has been extracted and is now included with new versions of ROM files for games that require this information.

If you want to use your current ROM files, you can obtain just the PROM images in one large file at:

<http://mame.retrogames.com/> <<http://mame.retrogames.com/>>

3.17 Why is xname/xmess slow?

Xname/xmess is a compiled C program that is emulating hardware. Software emulation of hardware always incurs a speed penalty. Here are some things you can do:

- Try using the `-frameskip` command line parameter.
- Try pressing F8 while running xname/xmess to alter the frame rate.
- Try disabling the throttle with the `-nothrottle` command line parameter or by pressing F10 while running xname/xmess.
- While running xname/xmess, press F11 to check the speed of xname/xmess. If you never get a display of 100%, you need a faster computer.

You can also try recompiling xname/xmess with more optimal settings. The following suggestions may not work or may not be appropriate on some machines and in some installations. However, in cases where they do work, there can be a noticeable increase in overall performance of xname/xmess.

These "enhancements" simply require editing the Makefile that is distributed with xname, and then recompiling the source. If you are compiling the source for the second time you will need to do:

```
make clean
make
make install
```

after editing the Makefile to set your new settings.

3.17.1 SPEEDUP # 1 - Inline compiling

Try enabling inline compiling by commenting and uncommenting the following lines:

```
# IL      = '-DINLINE=static inline'
IL      = -DINLINE=static
```

3.17.2 SPEEDUP # 2 - Direct Graphics Access

For a big boost, try enabling DGA by uncommenting the following line:

```
DGA = 1
```

Then start xname as root, and press left-alt + home to switch to dga mode; that should be a lot faster, and doesn't need/use xsync. ;)

To get back to a window without exiting, press left-alt + insert.

The benefit of DGA is that it is Direct Graphics Access and therefore writes directly to the framebuffer (think DirectX versus GDI under MS Windows). So it should be a tad faster, unless you've got a really really well accelerated X-server, which could be just as fast. Try pressing F11 to see the actual framerate, although it might be hardly noticeable (visually, at least), it should give you some speed increase.

Besides that, it automatically switches to the right mode and removes any window manager stuff, but that can be done without DGA too. (I might add such a feature later, since it allows easy fullscreen without running as root.)

Adding a 320x240 mode will most definitely get you a very nice fullscreen display at scale = 1. ;)

Don't forget that you still need to set up the Modes line of the "Display" subsection of the "Screen" section in your X11 config file if you want the games to fill the screen in DGA mode. Xname will pick the best mode it can find, so if you have a line like

```
Modes "1024x768" "800x600" "640x480" "320x240" "256x224"
```

then any game which fits into a 320x240 screen can be displayed in that resolution, and so on. Don't forget your X11 config file might have a number of different "Screen" sections in it, and you will need to add Modes to the one that is actually active with your Xserver is running. And of course it's not always that easy, since you also need to tell X which monitor settings are required for these resolution modes. This is done through Modeline entries in the "Monitor" section of your X config file.

You can check out the `/contrib/tools/modelines` file for a set of crazy modes made especially for xname in DGA mode. They should work on any reasonable multisync monitor, but don't use them on an old fixed-frequency monitor. (And if you can't resist trying, I won't pay for a new one!)

3.17.3 SPEEDUP # 3 - NASM

You can also try installing nasm and compiling the asm core, that helps a bit too.

3.17.4 SPEEDUP # 4 - ICC

If you are running on an intel architecture machine, you could try to compile xname with the Intel C Compiler. Some people have reported speed gains from +10%.

3.18 I've got a dual processor machine. What can I do to take full advantage of this?

Not much. Xname is a single-threaded port of a single-threaded DOS app. Some degree of speedup will hopefully be found on an SMP machine since xname is more likely to get to hog its one CPU for longer. But of course the same applies to any CPU-intensive program. And all the things one would normally do to any application to give it more CPU time (higher priority, kill or renice unnecessary process, etc.) will work with xname.

Now you may at this point be thinking, "Surely you could have one thread doing emulation and another thread doing graphics to get a speedup." Yes, but there are some serious issues to consider. The first is synchronisation between the threads, which needs to be pretty tight. If not done smoothly, the overhead will blow most of the benefit away. Also, it needs to be implemented with a portable threads interface ... which basically means POSIX threads. In a nutshell, it's a lot of effort for probably not much gain. But of course if you are feeling particularly energetic and in the mood to code, then patches can always be submitted to the xname devel mailing list ... :)

3.19 Why doesn't my joystick handle diagonal movement correctly on my Intel x86 system?

Xname/xmess uses autocalibrating, but you could try to use specific programs to calibrate it (jscale that comes with joystick module works fine...).

3.20 How do I enable XinputExtensions X11 based joystick in Xfree86?

You need to edit XF86Config to enable the XInput section. See the manual page for XF86Config.

3.21 How do I read a manual page?

Easy. If you want to read the manual page for XF86Config, type: man XF86Config

3.22 Why do I get gz_open symbol not found while compiling?

Xname/xmess versions newer than 0.35 use zlib for some compression routines. Unfortunately, a lot of Linux distributions have an old version of zlib installed as /usr/X11R6/lib/libz.a, and the new version installed as

/usr/lib/libz.a. If this is the case, it's harmless to remove /usr/X11R6/lib/libz.a. If you're not sure whether this is the case, just move /usr/X11R6/lib/libz.a to another location.

Removing or just moving /usr/X11R6/lib/libz.a should fix this problem.

Also see: 3.23 (FAQ: Why do I get zlib.h not found?)

3.23 Why do I get zlib.h not found?

Xmame/xmess versions newer than 0.35 use zlib for some compression routines.

Zlib is standard on most Unix versions. For those who don't ship with zlib or ship with a version that is too old, xmame/xmess now comes with a stripped-down version of zlib. Uncomment the appropriate line in the Makefile to use this.

3.24 When I run xmame.x11, nothing happens—no window pops up or anything. Help!

Are you running a recent version of KDE, such as 2.1.1? It appears that there's a bug with KDE's window manager that prevents some X11 apps from popping up their main window. You can either run the SDL version of xmame, or try the following workaround, which several people have reported as being successful:

```
xmame.x11 -rid 'xwininfo -root | grep xwininfo: | awk '{print $4}'
```

3.25 Can I run xmame fullscreen?

Yes, of course you can. There are several ways to accomplish this, each with its own advantages and disadvantages.

To enable the fullscreen support, you will need to change some settings in the makefile and recompile xmame.

Besides the x11 and SDL drivers, the FX and GL drivers are also capable of playing games full screen. There is a separate chapter for these later on.

3.25.1 xmame.x11

First, before recompiling, make sure you have an xserver with the "xfree-dga" extension enabled (do a "xdpyinfo" and see if you've got the "xfree-dga" extension); any recent release of XFree86 should have it.

Select DISPLAY_METHOD = x11 and uncomment the line "DGA=1" in the makefile.

Now build xmame and give the xmame.x11 file permanent root permissions so that you don't need to start xmame as root (do a "chown root xmame.x11" and a "chmod u+s xmame.x11").

If you want to start immediately in fullscreen mode, start xmame.x11 with the "-x11 1" argument or specify "x11-mode 1" in your xmamerc file.

During play you can switch between windowed and fullscreen mode. See 3.12 (What keys does xmame/xmess use?)

3.25.2 xmame.sdl

The SDL library was meant to make it easy for programmers to make full screen applications. To start SDL in fullscreen mode, just run `xmame.SDL` with the option `'-fullscreen'`. If you want to know the resolutions that SDL can use, run `"xmame.SDL -listmodes"`, choose the resolution you want, and start `"xmame.SDL -modenumber <insert chosen number here>"`.

You can add `"fullscreen 1"` to your `xmamerc` file if you want SDL to start fullscreen automatically; the `modenumber` line can also be used in the `xmamerc`. :-)

Most problems experienced by people who are trying to use SDL for fullscreen are related to the section 3.26 (fullscreen has a large border)

During play you can switch between windowed and fullscreen mode. See 3.12 (What keys does `xmame/xmess` use?)

3.26 xmame.SDL / fullscreen has a large border around the game.

If the game runs in fullscreen but occupies only a small area in the center of the screen, then your `xserver` does not have access to low resolutions (320x240, 400x300, and 512x384, for instance). In the case of SDL, the window will be in the resolution of closest matching resolution that SDL can switch to for full screen display.

In this case, you have to add `"Modelines"` for these resolutions in your `XF86config` file and add these resolutions to the resolutions list of your display somewhere in your `XF86config` file (you know, the list of resolutions you can switch to using `ctrl+`).

You should find `modelines` that works with most graphic cards in the `xmame contrib` subdirectory.

If you don't want to alter your `XF86config` file, another solution is to run `xmame.x11` with the `"-scale 2"` argument. In this case, the emulator will use 640x480 instead of 320x240, 800x600 instead of 400x300, etc. But I don't recommend it. The scaling uses `cpu` and the games will run more slowly.

However, if your `cpu` is fast enough, a better option is to use the effects `"-ef 1"`. This introduces advanced render methods to create a nicer looking effect than the simple scaling. On the website you can see *screenshots* <<http://x.mame.net/gamescreenshots.html>> of the different effects.

3.27 I only use some of the emulation parts (Neo Geo / CPS-2 / ...). Do I need the entire xmame?

If you are interested in compiling only some of the games you, can compile `xmame` with an alternate target. Edit the `makefile.unix` and change the target line to accomplish the desired effect. Using this approach, only the CPUs, sound devices, and other chips needed for that particular target are compiled, resulting in a much smaller executable.

- `'TARGET = cpmame'` makes a `mame` version that will only emulate the CPS-1 and CPS-2 games
- `'TARGET = neomame'` makes a `mame` version that will only emulate the Neo Geo system

There are some other targets but they are really only useful for the `mame` developers who create new drivers.

3.28 Why does xmame freeze and lock up when I hit <ESC> to exit the game or P to pause the game?

There appears to be a bug in the "timer based audio" code that xmame/oss uses on at least one motherboard that we are aware of. Specifically, the ATX 845Pro (Intel Pentium 4 processor) with an integrated i810 soundchip.

To get around this bug use Esound (Enlightened Sound Daemon), found as esd with Gnome distributions, instead of the oss sound drivers.

Compile ESOUND support into the xmame executable by uncommenting the appropriate line in makefile.unix. Tell xmame to use esound by specifying "esound" on the "dsp-plugin" line in xmamerc, or use the -dp command line parameter to select esound. Also see the next section below.

3.29 Why does xmame report "can't resolve host name '/dev/dsp' when trying to use esound?"

Remove the words "/dev/dsp" from the "audiodevice" line in xmamerc. In other words, you want to have the word "audiodevice" on a line all by itself, with no audiodevice specified.

3.30 Where can I get the latest version of xmame/xmess?

The xmame/xmess Home Page always has the most up-to-date version. The address is

<http://x.mame.net/> <<http://x.mame.net/>>

3.31 I upgraded xmame and now some games do not work anymore.

This can happen, especially if the drivers are in full development or some internal core routines have changed and the drivers have to be updated to accomodate for these core changes. Sometimes a new romdump has been made from the game and you still have the old version, or the zipfile has changed names and you need to rename yours (or any combination of these factors :-).

Here are some things you can do:

1. Clear all the old configuration files, nvram files, etc. Also delete the content of the directories mem, nvram, cfg, and try to run the game again. Don't forget to delete the default.cfg file!
2. If that fails, check to see if the name of the zipfile is still in the list of supported games. If you enter an invalid game name, xmame will take a guess to determine which game you wanted; look at the first lines of output to see if it tries to load a different game than the one you want. To know if your game is still supported, try 'xmame.x11 -list gamename'. If you do not pass a game name, the complete list of supported games is printed, and you can search for the new game name (better use 'xmame.x11 -listfull').
3. If the name hasn't changed, check to see if you have the correct romdump. Run 'xmame.x11 -verifyroms gamename' to check your romset. If this fails you will need to download a newer romdump. Also check your parent roms as a possible cause of this romset failure.

4. Read the documentation!! Or like the experts would say, "RTFM!" :-) Chances are that if the driver is broken, the developer has noted this in the documentation.
5. If the above steps don't solve the problem, then check to see if it is a known problem; if necessary, report it to the mame testers project.

If none of these fixes the problem, then try using the older version of xmame you have backed up somewhere to play the game. (You did back-up, didn't you?)

3.32 Where can I send fixes, bug reports, suggestions, or money? :-)

See 5.1 (Mail) of this document.

3.33 What if my question is still not answered?

Please check the official mame and mess readme files to see if that answers your question:

- *The official mame readme file (readme.mame)* <readme.mame>
- *The official mess readme file (readme.mess)* <readme.mess>

If that doesn't help, then ask your question on the mailing list or send it to the xmame/xmess maintainer, see 5.1 (Mail) for contact information.

4 Display Subsystem-Specific Comments and Notes

4.1 FXmame, a glide driver for xmame

The latest version of this driver can be obtained at <http://glmame.linuxgames.com> <<http://glmame.linuxgames.com>>.

4.1.1 Requirements

To run FXmame, you must have:

- A 3Dfx card
- Glide 2x
- The xmame distribution

4.1.2 Compiling and Installing

FXmame comes in two flavours—`xfx` and `svgafx`. The `xfx` version is designed to run under X11. It uses an X window for input. The `svgafx` version runs from the console, and uses `svgalib` for input. Both use the 3Dfx hardware for rendering.

To install FXmame, follow the general instructions for installing `xmame`. To specify the glide driver, do the following to the Makefile:

- Select either `"xfx"` or `"svgafx"` as your display method
- Configure the glide variables to correspond to your glide setup

The FXmame executables are called `"xmame.xfx"` and `"xmame.svgafx"`

4.1.3 Running

FXmame runs with the same parameters and uses the same keyboard commands that `xmame.x11` does.

A few extra keyboard commands have been added in the glide driver:

- `RIGHTCTRL-A` – toggle antialiasing in vector games
- `RIGHTCTRL-B` – toggle bilinear filtering

The glide driver also adds several command-line options:

```
-fxgama <value> 3Dfx gamma correction value
```

Note that in order to display the game options or the fps display in vector games, I have to overlay the game bitmap over the vector display. This is because I circumvent the bitmap entirely in vector games. Drawing the bitmap is a big cpu hog, so you'll notice that the game slows down when you turn on the fps display (I know, not ideal when you want to find out the frame rate...).

4.1.4 Common Questions (FAQ)

Q: When I run FXmame, it just dumps core.

A: Make sure you are running FXmame as root. Root privileges are required to access the 3Dfx hardware

Q: When I run `xmame.xfx`, it just hangs at the initial game screen.

A: Are you running KDE? For some reason, I can't get keyboard input under KDE. If you have this problem, switch window managers or use `xmame.svgafx` instead.

4.1.5 Present Limitations/Bugs

- Antialiasing isn't working quite as well as it should
- Input problems under KDE

4.1.6 Future Work

- Optimize the drawing code to update only the changed portions of the game bitmap.

4.1.7 License

The FXmame code is Copyright 1998 by Mike Oliphant. It may be used and distributed under the terms of the MAME license.

4.1.8 Version History

v0.5 – December 11th, 1998 —————

- implemented long copies in screen update
- added dirty handling. this greatly increases the speed of games that support the dirty update strategy (although most of them are older games that already ran at full speed on modern hardware)
- merged source with xmame b8.1

v0.4 – October 30th, 1998 —————

- improved alpha handling in vector games - now bitmapped overlays do not dim the screen
- added svgalib support (xmame.svgafx)
- fixed a tiling bug that gave artifacts on texture tile boundaries

v0.3 – October 29th, 1998 —————

- got rid of extra lines in vector games
- added bitmap overlay in vector games to show options/fps
- added antialiasing in vector games

v0.2 – October 28th, 1998 —————

- added screenshots
- added 3Dfx gamma adjustment
- added direct acceleration of vector games (still buggy)

v0.1 – October 27th, 1998 —————

- first public release

4.1.9 Author

Mike Oliphant (oliphant@ling.ed.ac.uk) <<mailto:oliphant@ling.ed.ac.uk>>
<http://glmame.linuxgames.com> <<http://glmame.linuxgames.com>>

4.2 GLmame, an OpenGL driver for xmame

The latest version of this driver can be obtained at: <http://www.jausoft.com/glmame.html> <<http://www.jausoft.com/glmame.html>>

4.2.1 Requirements

To run GLmame you must have:

- An OpenGL implementation
- libjpeg
- hardware 3D acceleration (if you want any kind of framerate)
- the xmame distribution

4.2.2 Compiling and Installing

Follow the general instructions for installing xmame. To specify the OpenGL driver, do the following in the Makefile:

- Select "xgl" as your display method.
- For runtime printed debug info, add "-DGLDEBUG" to the "GLCFLAGS" variable.
- GLU version 1.2 is recommended (e.g., from SGI). If you use Mesa's GLU, you may have to add "-DGLU_VERSION_1_2" to the "GLCFLAGS".

The GLmame executable is called "xmame.xgl".

4.2.3 Running

xmame.xgl runs with the same parameters and uses the same keyboard commands that xmame.x11 does.

The program starts up in cabinet mode. While this is pretty, and cool, you'll probably want to actually play in fullscreen mode.

A few extra keyboard commands have been added in the OpenGL driver:

- RIGHT_ALT-A – toggle antialias filtering (vector)
- RIGHT_ALT-B – toggle bilinear filtering (raster)
- RIGHT_ALT-C – toggle cabinet/fullscreen display (both)
- RIGHT_ALT-F – toggle between fullscreen (max window size) and startup window size (both)
- RIGHT_ALT-O – toggle draw-bitmap (both; nonsense for raster games, speedup for vector games)
- RIGHT_ALT-T – toggle phosphor trails (vector)
- RIGHT_ALT-PLUS_PAD – incr. beam size (vector)
- RIGHT_ALT-MINUS_PAD – decr. beam size (vector)

The OpenGL driver also adds several command-line options:

-[no]fullscreen	Start in fullscreen mode (default: disabled)
-[no]gldblbuffer	Disable/enable double buffering (default: enabled)
-gltexture_size <int>	Force the max width and height of one texture segment (default: autosize)
-[no]glforceblitmode / -[no]glblit	Force blitter for true color modes 15/32bpp (default: enabled)
-[no]gltext78 / -[no]gltext	Force the usage of GL extension number 78, if available (palletted texture) (default: enabled)
-[no]glbilinear / -[no]glbilin	Disable/enable bilinear filtering (default: enabled)
-[no]gldrawbitmap / -[no]glbitmap	Disable/enable drawing the bitmap; e.g., disable within vector games for a speedup (default: enabled)
-[no]glcolormod -[no]glcmod	Disable/enable color modulation (intensity, gamma) (default: enabled)
-glbeam <float>	Set the beam size for vector games (default: 1.0)
-[no]glalphablending / -[no]glalpha	Disable/enable alpha blending if available (default: enabled)
-[no]glantialias / -[no]glaa	Disable/enable antialiasing (default: enabled)
-gllibname / -gllib <string>	Choose the dynamically loaded OpenGL library (default: libGL.so)
-glulibname / -glulib <string>	Choose the dynamically loaded GLU library (default: libGLU.so)
-[no]cabview	Don't start/start in cabinet view mode (default: don't start in this mode)
-cabinet <cabname>	Specify which cabinet model to use

```

                                (default: glmamejau)

-glres <string>                Always scale games to <Xres>x<Yres>,
                                keeping their aspect ratio.
                                This overrides the scale options.
                                Use this e.g. for the Vodoo1 gfx cards
                                with: "-glres 640x480"

```

Three cabinet models are provided. The default, "glmamejau", is designed to look like a standard arcade cabinet, like "glmame". The last one, "trans", is a box with a transparent game screen.

Note that in order to display the game options or the fps display in vector games, the game's bitmap is overlaid above the vector display. Drawing the bitmap is a big cpu hog for some low-end hardware accelerated OpenGL implementations, so you'll notice that the game slows down when the bitmap is overlaid.

Because bitmap overlaying is the default, you can toggle it on and off by pressing the "RIGHT_ALT-O" key, or turn it off at startup by using the command line option "-noglbitmap".

A problem with vector games and the cabinet mode is that some vector points do not reside within the screen coordinates, so you may see some vectors outside of the cabinet's name display.

*** IMPORTANT! ***

If you have trouble with GLmame, or it is slow, read the "Common Questions" section. Most of the questions people email me are answered here.

4.2.4 Cabinet Models

Cabinet model information is stored in the "cab" subdirectory of the directory you specified with the ROMPATH variable in the Makefile. Path information specified in xmamerc or with the 'rompath' shell environment variable is currently not used. Each cabinet model has it's own subdirectory. GLmame will first try to load either the user-specified cabinet model (using the -cabinet command-line option), or a model corresponding to the current game name. Failing that, it will use the generic "glmame" model.

A cabinet model directory must contain a geometry file <modelname>.cab, and any .jpg files used for textures on the model. The format of the .cab file will be familiar if you have ever programmed using OpenGL – it looks much like an OpenGL display list. The format is as follows:

```
cabv1.1
```

This header must be the first 7 characters of the file.

```
#<text>
```

Comment lines begin with '#'

```
begin <type>
```

Begins a geometry object, where type is one of (points, polygon, quads, quad_strip, screen). Each "begin" must be paired with an "end" statement. A "polygon" is specified using one vertex per, well... vertex. "quads" are four-sided polygons, each specified by four vertices in order around the polygon. A "quad_strip" is a series of four-sided polygons, each of which shares an edge with the polygons before and after it in the strip. A "screen" is a quad that defines the actual area in 3D space where the game will be displayed. only one of these should be defined. For more info, look at any description of OpenGL geometry.

```
end
```

Ends the geometry object begun with "begin".

```
pointsize <s>
```

Changes the current pointsize to the corresponding size value (specified as a floating point number between 0.0 and, e.g., 20.0). Note: You cannot use this statement between begin and end!

```
color4 <r> <g> <b>
```

Changes the current color to the corresponding rgb value (specified as floating point numbers between 0 and 1).

```
color4 <r> <g> <b> <a>
```

Changes the current color to the corresponding rgba value (specified as floating point numbers between 0 and 1).

```
shading <type>
```

Sets the shading type. available types are "smooth" (which blends colors from one vertex to the next) or "flat" (which doesn't).

```
vertex <x> <y> <z>
```

Specifies a vertex for the current geometry object as a point in 3-space.

```
numtex <num>
```

Specifies the number of different textures to be loaded. This declaration must precede any "loadtex" call.

```
loadtex <num> <width> <height> <filename>
```

Loads texture number <num> from .jpg file <filename>. The .jpg file *must* be <width> by <height> pixels, and both <width> and <height> must be powers of two (and must conform to the hardware requirements of any 3D card you want the cabinet to work with – 256x256 for 3Dfx).

```
texcoord <xcoord> <ycoord>
```

Sets the texture coordinate of the next vertex. This specifies where in 2D texture space the vertex lies.

```
enable texture
```

Enables texture mapping (using the currently-selected texture).

```
disable texture
```

Turns off texture mapping.

```
settex <num>
```

Select texture <num> for doing texturing. The texture must previously have been loaded using "loadtex".

```
camerapan <num>
```

Starts the camera pan definition that will be followed by <num> camera transistions. After the last transision definition, the pan sequence must be ended with an "end" statement.

```
goto <lx> <ly> <lz> <px> <py> <pz> <nx> <ny> <nz>
```

Relocates and reorients the camera. The arguments are the same as are used by the OpenGL command "gluLookAt()". The camera is positioned at point <lx,ly,lz>, pointing along vector <px,py,pz>, with "up" normal (the direction that is "up" for the camera) <nx,ny,nz>.

```
moveto <lx> <ly> <lz> <px> <py> <pz> <nx> <ny> <nz> <frames>
```

Moves the camera to a new location (specified as in "goto"). The transition occurs over <frames> frames.

All commands must be on separate lines. Use the (somewhat) commented glnamejau cabinet as an example to work from.

4.2.5 Common Questions (FAQ)

Q: I am only getting a few frames-per-second. This sucks! What gives?

A1: You're probably running Mesa in software mode. You should read the "README.3Dfx" file in the Mesa distro. Specifically, you need to "setenv MESA_GLX_FX fullscreen".

A2: You don't have a 3D graphics card. Go buy one (see below)!

Q: When I run xmame.xgl, it just dumps core. That's not as much fun as 3D Pacman, now is it?

A1: If you are using a 3Dfx card, make sure you are running xmame.xgl as root. Root privileges are required to access the 3Dfx hardware.

A2: If you're using Mesa and haven't upgraded to v3.0, do so.

Q: When I run xmame.xgl, it just hangs at the initial game screen.

A: Are you running KDE? For some reason, I can't get keyboard input under KDE. If you have this problem, switch window managers.

Q: Which graphics cards are usable with `xmame.xgl`?

A: We have the following experiences (date 20010713):

- Voodoo3 + DRI OpenGL driver (XFree86 4.0.2), with command line options `"-noglaa"` and 16bpp X11 server color depth.
- Voodoo1 + Mesa 3.4 + Glide, with command line options `"-glres 640x480"`!
- GeForce + NVIDIA OpenGL driver + XFree86 4.0.2 with default command line options and 16/24bpp X11 server color depth.
- NVIDIA TNT2 + NVIDIA OpenGL driver + XFree86 4.0.2 with command line options `"-bpp 16"` and 16bpp (may be 24bpp also ???) X11 server color depth (OpenGL colortable mapping is buggy).
- NVIDIA TNT + NVIDIA OpenGL driver + XFree86 4.0.2 with command line options `"-bpp 16"` and 16bpp (may be 24bpp also ???) X11 server color depth (OpenGL colortable mapping is buggy).
- Matrox MGA 400 + DRI OpenGL driver (XFree86 4.0.2) with command line options `"-bpp 16"`, `"-noglaa"` (???) and 16bpp (may be 24bpp also ???) X11 server color depth (OpenGL colortable mapping is buggy).
- ATI R128 + DRI OpenGL driver (XFree86 4.1.0) with command line options `"-noglex78 -noglaa"` or `"-bpp 16 -noglaa"`, where the latter options are a bit slower (about 5 fps).
- LinuxPPC - Powerbook-Firewire + ATI R128: see above!

All of the above drivers work with the option `"-bpp 16"` and/or `"-noglex78"`, the latter of which is very slow, of course! So it looks like an OpenGL implementation bug is causing these driver/hardware combinations to only work with these command line options. The bug may lie within these OpenGL functions:

- `glColorTableEXT`
- `glColorSubTableEXT`

Currently fast and funny driver/hardware combinations are:

- NVIDIA GeForce + NVIDIA OpenGL driver + XFree86 4.0.2; I have tested the GeForce256 (AGP) hardware.
- Voodoo + DRI OpenGL driver (XFree86 4.0.2); I have tested the Voodoo3 hardware.

Q: When I run `xmame.xgl`, everything is messed up.

A: Read the above experiences ... and try

- switching your X server's bpp value, e.g. switch from 32bpp or 24bpp to 16bpp;
- the command line option `"-bpp 16"`;
- the command line option `"-noglaa"`;
- the command line option `"-noglex"`.

4.2.6 Present Limitations/Bugs

- No clipping of vector games (most noticeable in cabinet mode).
- No preservation of game aspect ratio in cabinet mode.

4.2.7 Future Work

- Fixing existing bugs .. are there any ? Or are there some unknown features only ;-) ?
- Optimize the drawing code to update only the changed portions of the game bitmap. Optimizing the dirty code ..

4.2.8 License

The GLmame code is based upon Mike Oliphant's GLmame code version 0.6.

The GLMame code is Copyright 1998 by Mike Oliphant. It may be used and distributed under the terms of the MAME license.

Since GLmame version 0.7, the code has been changed and improved heavily by Sven Goethel — no license change, of course!

4.2.9 Version History

v0.83 – March 29th, 2001 —————

- Changed the code heavily to improve compatibility and speed. The new code is a re-merge of the GLMame32 code, which is a branch of Mike Oliphant's GLmame code version 0.6.
- For a detailed list, please consult the file changes.opengl within the distribution's documentation directory! You can also view the current changes.opengl here: <http://www.jausoft.com/glmame.html> <<http://www.jausoft.com/glmame.html>>

v0.6 – December 14th, 1998 —————

- added support for systems that do not support paletted textures (SGI and solaris OpenGL implementations, for example)
- merged source with xmame b8.1

v0.5 – October 30th, 1998 —————

- fixed a bug with texturing being left on in bad places
- added 3Dfx gamma adjustment

v0.4 – October 21st, 1998 —————

- updated to work with xmame v0.34b5.2

v0.3 – October 12th, 1998 —————

- fixed verticle screen centering in fullscreen mode
- moved cabinet and camera panning information into a user-editable configuration file
- added 'trans' cabinet model o added double buffer, view type, and cabinet selection command-line options

v0.2 – October 6th, 1998 —————

- adding missing 'cab' directory

v0.1 – October 5th, 1998 —————

- first public release

4.2.10 Authors

Since GLmame version 0.7: Sven Goethel (*sgoethel@jausoft.com*) <mailto:sgoethel@jausoft.com>
<http://www.jausoft.com/glmame.html> <http://www.jausoft.com/glmame.html>

Till GLmame version 0.6: Mike Oliphant (*oliphant@ling.ed.ac.uk*) <mailto:oliphant@ling.ed.ac.uk>
<http://glmame.linuxgames.com> <http://glmame.linuxgames.com>

5 Contact Information and Credits

5.1 Mail

There is an official xmame/xmess mailing list. All operations related to this list are performed via email to *Xmame-request@toybox.twisted.org.uk* <mailto:Xmame-request@toybox.twisted.org.uk>

- To subscribe, send mail to: *Xmame-request@toybox.twisted.org.uk* <mailto:Xmame-request@toybox.twisted.org.uk> with the keyword "subscribe" in the body.
- To unsubscribe, send a mail to: *Xmame-request@toybox.twisted.org.uk* <mailto:Xmame-request@toybox.twisted.org.uk> with the keyword "unsubscribe" in the body.
- To change your subscription, get a password reminder, etc., go to <http://toybox.twisted.org.uk/mailman/listinfo/xmame> <http://toybox.twisted.org.uk/mailman/listinfo/xmame>.
- To send mail to the list, use: *xmame@toybox.twisted.org.uk* <mailto:xmame@toybox.twisted.org.uk>
- To ask specific questions about the xmame mailing list, send mail to: *Xmame-request@toybox.twisted.org.uk* <mailto:Xmame-request@toybox.twisted.org.uk>

If you're just curious, and only want to take a look into the mailing list without subscribing, you can use the web-based archive at <http://www.mail-archive.com/index.php3?hunt=xmame> <<http://www.mail-archive.com/index.php3?hunt=xmame>> or <http://toybox.twisted.org.uk/pipermail/xmame/> <<http://toybox.twisted.org.uk/pipermail/xmame/>>. (The latter was activated on 5 January 2002.)

To ask UNIX-specific xmame/xmess questions, you could send mail to: *Lawrence's folder* <<mailto:gold@aros.net>>, but you're better off sending mail to the list.

To ask general questions about MAME, send mail to: Mirko (Mix) Buffoni *Mirko (Mix) Buffoni* <<mailto:mix@lim.dsi.unimi.it>>

Also see (and post) articles on USENET group *comp.emulators.misc* <<news:comp.emulators.misc>>

with a proper keyword in the body.

Other proper keywords currently recognized as commands by *Xmame-request@toybox.twisted.org.uk* <<mailto:Xmame-request@toybox.twisted.org.uk>> are:

```

who           - see everyone who is on the list
info          - view the introductory information for this list
help          - list all recognized commands

```

Xmame Mailing List

- The xmame mailing list is intended to be a technical forum for xmame developers, that is,
 - To exchange ideas for producing a better xmame.
 - To share patches and bug notifications.
 - To study new design strategies.
 - To expose problems and learn know-how.
- The Xmame mailing list **is not** a forum about playing games (but everyone on the list plays...); so please
 - **Don't use the list to ask about ROMs.**
 - **Don't ask about non-UNIX ports of MAME, just about MAME and xmame.**
 - **Don't ask about implementing a game. Just search the docs and contribute yourself (you'll enjoy twice: programming and playing...).**
 - **Don't show your last high score in xxx game (we are interested in programming, and then playing, in this order).**
- **Copyright notice: NEVER, I repeat NEVER send ROM images to the list.** (But you can tell us where to retrieve them... ;-))
- At this moment, the xmame list is **unmoderated**; that is, everyone subscribed will receive everything sent to the list. We foresee a need to change this policy since the list starts to suffer from spam :-).

If you want to send a personal mail to the xmame/xmess coordinator, send to the address below. Everyone who contributes to xmame/xmess should subscribe to the xmame/xmess mailing list.

- Mailing list: *Xmame-request@toybox.twisted.org.uk* <mailto:Xmame-request@toybox.twisted.org.uk>
- Xmame/xmess coordinator: *gold@aros.net* <mailto:gold@aros.net>

That's all. Enjoy xmame!

5.2 Bugzilla

In addition to/instead of posting bugs to the mailing list, you can report them to the MESS Bugzilla database at <http://bugzilla.mess.org/> <<http://bugzilla.mess.org/>>, under the "XMAME/XMESS" category.

5.3 Credits

Lots and lots and lots of thanks to everyone for their great help on the xmame project. I would like to thank the following people and of course anyone I've forgotten.

Thank you all,

Hans

(Addendum: I'd like to send a big thanks to Hans for his fantastic work as longtime xmame/xmess maintainer, and also to those on the mailing list who have been most helpful during my transition to official maintainer. –Lawrence Gold)

1. Fathers of the mame-creature (main version)
 - Nicola Salmoria
 - Allard van der Bas
 - Mirko (Mix) Buffoni (General Mantainer)
2. Fathers of mess
 - Brad Oliver
 - Richard Bannister
 - Ben Bruscella (current coordinator)
3. Original UNIX/X11 port
 - Allard van der Bas
 - Dick the Ridder
 - Juan Antonio Martinez
4. Semi-retired Unix maintainer extraordinaire
 - Hans de Goede
5. Alpha-specific issues
 - Christian Groessler

6. Sun-specific issues

- Keith Hargrove
- Mathis Rosenhau

7. Irix-specific issues

- Tristram Scott

8. QNX-specific issues

- Travis Coady

9. GGI maintainance

- Gabriele Boccone (original port)
- Christian Groessler (current maintainer)

10. OpenGL code & maintainance

- Mike Oliphant (original creator)
- Sven Goethel (from GLmame 0.7 to the present)

11. Network code

- Eric Totel

12. Perl scripts for automation of some porting issues

- Bill Adams

13. Xmame RPMs

- Jeremy Hansen

14. Mailing list maintainer

- Chris McCraw (retired)
- Pete French

15. Bits and pieces

- Christian Groessler
- Torsten Paul
- Jack Patton
- David Black

16. AIX Sound code

- Chris Sharpp

17. IRIX Sound code

- *entropy@zippy.bernstein.com* <mailto:entropy@zippy.bernstein.com>

18. German readme
 - Robert Hamberger
19. Documentation (SGML, HTML, MAN)
 - Rene Herrmann
20. Bugzilla bug database
 - Sean Young

6 Legal Issues and Copyrights Concerning MAME/MESS

- **Please read the MAME/MESS readme-files first!**
- Xname/xmess should be distributed under terms of the license in *readme.mame* <readme.mame>
- And if you are interested in mess, take a look in *readme.mess* <readme.mess>
- The main issue involves ROM images. ROM images are protected under the copyrights of their authors, and CANNOT be distributed in the same package as xname/xmess. Download any ROM images at your own risk and responsibility.
- XInputExtensions-based Joystick support. Source code is entirely pure hack work. No sample code, no info at all was found (only library reference manual) and is copyrighted by me (Juan Antonio Martinez). Use it under terms of GNU General Public License.

7 Some Links

7.1 Unix sources and docs

- The xname/xmess sources *xmess-0.561.1 sources* <download/xmess-0.561.1.tar.bz2>
xname-0.60.1 sources <download/xname-0.60.1.tar.bz2>
- xname/xmess FAQ & Installation Guide: *xname-doc* (pdf format)
- xname/xmess known bug list: *open issues*

7.2 Unix binaries

- *xname RPMs* <<http://www.forte-intl.com/~ronald/xname/>>
- *xname+CPS2 RPMs* <<http://www.beimborn.com/cps2/linux/rpms>>
- *Source and binary for Solaris UltraSPARC* <<http://come.to/georgie>>
- *AMD-optimized builds for various targets* <<http://kwyxz.nerim.net/klinux/xname.html>>

7.3 Xname/xmess download mirrors

- <ftp://cps2:cps2@wingnut.beimborn.com> <<ftp://cps2:cps2@wingnut.beimborn.com>>(US)
- <http://www.mame.dk> <<http://www.mame.dk>>(Denmark)
- <ftp://ftp.kaupp.cx/pub/xmame> <<ftp://ftp.kaupp.cx/pub/xmame>>(US)

7.4 (X)Mame front-ends

- *The official mame frontend page* <<http://www.mame.net/frontend.html>>
- *TkMame* <<http://tkmame.retrogames.com>>, the Tcl/TK front-end that works in Linux/Unix and MS Windows
- *Xmameload* <<http://www.toetag.com/~tom/xmame.html>>
- *Kmamerun* <<ftp://ftp.sad.it/pub/kde/local/>>
- *Mamecat* <<http://www.mameworld.net/mamecat/>>
- *GrokMame* <<http://user.cs.tu-berlin.de/~gator/mame/>>
- *Gmame* <<http://members.xoom.com/gmame/>>
- *startxmame* <<http://www.geocities.com/SiliconValley/Mouse/1570/>>
- *it-mame* <<http://it-mame.sourceforge.net/index2.html>>
- *gRustibus* <<http://grustibus.sourceforge.net>>, the **very good** GNOME front-end
- *IQF* <<http://www.sourceforge.net/projects/iqf/>>
- *GnoMame* <<http://gnomame.sourceforge.net/>>, another great GNOME-based front-end
- *LANZAMAME* <<http://lanzamame.mameworld.net/>>
- *Victor Manuel's TCL/TK Frontend* <http://www.terra.es/personal3/victorote/frontend/index_en.html>
- *ptkmame* <<http://ant.eelinux.com/ptkmame/>>
- *Kemulator* <<http://kemulator.sourceforge.net/>>, a KDE front-end with MESS and Stella support
- *Flynn* <<http://www.foxbox.org/adam/code/flynn/index.html>>(primarily for converted arcade cabinets)
- *Playmame & Playmess* <<http://www.swentelomania.be/swentel/module.php?boxcontent=35=40>> (perl-based menu makers)
- *jMame* <<http://www.voicenet.com/~jceklosk/>>, a Java-based front-end (requires JDE 1.4.0)
- *GXMame* <<http://gxmame.sourceforge.net/>>, a GTK-based front-end striving for MAME32 compatibility
- *CXMame* <<http://yeeking.net/cxmame/>>, a simple console-based front-end

7.5 Some other links

- The official M.A.M.E home is at <http://www.mame.net> <<http://www.mame.net>>
- The official mess site is <http://mess.emuverse.com> <<http://mess.emuverse.com>>
- Some other emulation sites are <http://www.vg-network.com> <<http://www.vg-network.com>> and <http://www.retrogames.com> <<http://www.retrogames.com>>
- The very best ROM site on this planet is at <http://www.mame.dk/> <<http://www.mame.dk/>>
- A couple of sites with CPS2 and other patches *Dan Beimborn's Linux CPS2Mame page* <<http://www.beimborn.com/cps2/linux/>> and *Stijn Hoop's xmame patches* <<http://www.win.tue.nl/~stijn/xmame/>>
- **MAME Action Replay Page** for those who want to have a look at other monitors at <http://marp.retrogames.com/> <<http://marp.retrogames.com/>>
- And, of course, don't forget our project homepage at <http://x.mame.net/> <<http://x.mame.net/>>

8 ChangeLogs

8.1 The MAME/MESS ChangeLogs

These are the changelogs of the official MAME and MESS release, the core of xmame/xmess.

- *The MAME changelog* (<changes.mame>) <<changes-mame.html>>
- *The MESS changelog* (<changes.mess>) <<changes.mess>>

8.2 The Unix ChangeLog

And now here is the HISTORY of the xmame project (of course that's why you are here ;-)) ... René

- *The Unix port changelog* (<changes.unix>) <<changes-unix.html>>
- The OpenGL display driver changelog: <changes.opengl> <<changes.opengl>> or <http://www.jausoft.com/glmame.html> <<http://www.jausoft.com/glmame.html>>