Arcade heaven

Simon Goodwin plays with XMAME, the arcade machine emulator for Linux.

On the CD and DVD

Your coverdisc includes emulator sources and many of the utilities mentioned in the article.

This month we test *MAME*, the *Multiple Arcade Machine Emulator*. This ambitious project emulates coin-operated arcade games, rather than home consoles or general-purpose computers. It's not one emulator, but a collection of modules emulating arcade cabinets custom-made to run thousands of microprocessor-based arcade games.

MAME is just five years old, though its originator, Nicola Salmoria, had a solid reputation writing Amiga freeware a decade ago. More than 100 programmers have bolstered his effort, and *MAME* has eclipsed earlier multi-emulation efforts like *Laser*, *Replay* and *System16* (*LXF19*), which supported specific families of arcade game hardware.

MAME shares processor emulation code with computer emulators based around the same chips, augmented by routines to mimic hardware added specifically for arcade use. *MAME's* modular design takes advantage of the hardware shared between products, developers and manufacturers. Source for *XMAME*, the Unix version, is on our coverdisc, with binary packages for common Linux systems. *XMAME* is one of the biggest emulators around, totalling 40MB of source, plus the game ROMs needed to bring it to life. The source includes driver data for over 3000 different ROM sets, dating from 1975 to 2001.

Electronic arcades evolved from Pinball, the first coin-operated entertainment with no prizes other than extended play, and custom themes and backdrops to update old cabinets. *MAME* emulates all the greatest microprocessor-based arcade games, and new ROMs are periodically added. If you've a set of Deniam, Konami GX, Sun, Orca or Seibu ROMs in the back of your shed, the MAME team would like to hear from you. They are particularly seeking Sega System 16 and 24 ROMs.

The documentation totals a megabyte of text, HTML and *Latex* data. Key information about each ROM set is embedded in the 13.5MB *XMAME* executable, indicating the screen format, control keys and which chips to emulate. Tab calls up controller mappings and other useful information while a game runs.

Graphic examples

XMAME displays can use an X11 window; *SVGAlib* for Linux on IBM-based PCs; GGI or SDL; OpenGL and 3Dfx extensions for X. The *SVGAlib* version can also drive 3Dfx hardware directly.

XMame 0.57 is fussy about the bit depth of the X display. The stable Debian *0.36* package works happily in my preferred 24 bit mode, but the latest release gave a garbled display and periodic segfaults until I bodged XF86Config to specify 16 bits per pixel.

Even *Space Invaders* – basically mono with simple gel strips overlayed top and bottom – claimed to need 32770 colours, when four might actually be enough. It appears that a maintainer – possibly for the more prescriptive Windows code – has wired in a dependency which Linux users would be better off without.

Glide and OpenGL versions support bilinear filtering for neater textures, and hardware anti-aliasing for smooth vector displays. *FXmame* is the Glide 2 version for 3Dfx graphics cards. It can use gamma correction to match colours and intensities more closely, but only runs as root as the driver needs direct access to 3Dfx chips which only the superuser can gain.

GLmame is the OpenGL version, with portable 3D support through the *OpenGL* abstract interface to 3D hardware and *libjpeg* compression. It can even emulate phosphor trails across the screen of vector displays, though not the flicker as they neared their redraw limit. It works with MESA 3, even in software-only mode, though MESA 4 is more reliable and you really need 3D graphics hardware such as ATI, Matrox or NVidia for a playable frame-rate. *XFree86 version 4* and 16 bit colour are needed.

Unconstrained by Microsoft, arcade designers pick the ideal processors for each product, tracking the best in CPU architectures. *MAME* emulates many once-trendy chips, including Motorola's 6800, 6802, 6805 and 6809, Intel's 8038, 8039, 8041 and 8085, and Texas Instrument's CPUs and DSPs like the 9900, 320 and 340 series. Chuck Peddle's ubiquitous 6502 and Zilog's Z80, Z180 and Z8000 are also much used. *MAME* can emulate MIPS RISC processors like those used in recent Sony and Sega consoles, and obscurer parts like the 2100 DSP, a 6809 variant from Konami, the 6309 – a souped up 6809 with extra goodies like a DIV instruction – and NEC's V series, first based on 16 bit Intel x86's, later embellished as the V60.

CCPU – Seventies RISC

CCPU is not a CPU optimised for the C language, though the PDP-11 minicomputer, emulated here by *T11*, comes pretty close. *CCPU* emulates a ground-breaking 1978-vintage 12 bit RISC specially invented for a dozen Cinematronics vector games like *Ripoff*, *Tail Gunner*, *Spacewar* and *Star Castle*. Another game special is ASAP, the Atari Simplified Architecture Processor, a custom RISC used in the *BeatHead* game.

Interchangeable graphics, sound and processor emulation modules do much of the work, but there are specific machine, driver and video hardware code modules for many of the ROM sets. These glue together the generic component support with *ad hoc* code and hardware mapping information, reflecting the custom wiring in each production cabinet.

Audio modules emulate Yamaha FM and OPL synthesis, GI's AY8910 and Atari Pokey beepers, a subset of the Maplin catalogue stalwart Philips SAA1099, generic DAC and ADPCM outputs, and dozens more, including speech chips like the TI5220 and Votrax speech synth, familiar from Gauntlet and SpeakAndSpell and the semi-human burbling in *Q-Bert*.

Apparently the coders at erstwhile pinball specialists Gottlieb gave up trying to program the Votrax to produce intelligible words and fed it random numbers to get a more impressive babble, like an adenoidal human using an alien language. *Q-Bert* graphics were inspired by M.C Escher, and the working title of the game was *Snots and Boogers*!

Sega's ***Afterburner*** **made good use of dual 68000 processors and a Z80 sound controller.**

Making MAME

MAME is well documented and available ready-compiled for many systems. *XMAME* updates lag slightly behind new Windows versions, but usually only by a few days. You may have to wait longer for PPC, ARM and 68K binaries, or versions tailored for displays other than X or *SVGAlib*.

The latest version of *XMAME* early this year was *0.57*, pre-compiled in packages with the RPM update incompatible with many existing distros. Compiling from source is the best way to get code that's sure to run on your machine, tailored for your libraries and hardware, but it's a big job, especially on a slow machine; if your connection uses anything better than tins and string, a binary download is worth considering, if you can find one suitable. That might mean opting for an old version, but anything since version *0.3x* will run most of the classic ROMs.

The source RPM is over 6.5 megabytes long. The tarball is packed with *bz2* compression rather than the usual *gzip*, so the name is *xmame-0.57.1.tar.bz2* rather than *.tar.gz* – but the difference is well worth it, as the *bz2* tarball weighs in at about five and a third megs. Unpack it with **tar -xvjf**; the **j** takes the place of the **z** that would signal *gzip* uncompression.

The Makefile explicitly supports DEC Alpha, Motorola 68K, PPC, Sparc, HP-PA and MIPS RISC chips as well as Intel 32 and 64 bit processors. *XMAME* has few dependencies – it expects *GCC* but you won't get far on Linux without that, and although it depends on the *zlib* compression library you don't absolutely have to install it, as a stripped-down version is included in the *XMAME* archive – uncomment the line **ZLIB=1** in the Makefile to use that.

Other options are worth tweaking in the well-commented makefile. Unless your copy of *GCC* is buggy you should use the fourth set of **CFLAGS** on an x86, commenting out the rather unambitious first set, or the fifth set for Linux on a PPC Mac or Amiga. Before you make changes type **cp makefile.unix Makefile** so you can edit a copy and simply type **make** to build your tailored version.

Set an alternative **DISPLAY_METHOD** if the slow but portable X11 default is not to your linking. You must compile versions separately if you want more than one, but can specify the alternate display by tacking a parameter onto the make command, so **make DISPLAY_METHOD=svgalib** builds the metal-bashing PC variant.

The **MASM_DEBUG** symbol requests *MAME's* built-in debuggers, but these retard emulation so consider if you really need them, typically to hack a game or implement a new ROM set. **MAME_NET** selects experimental multi-player support code, using UDP messages to sync machines across a network.

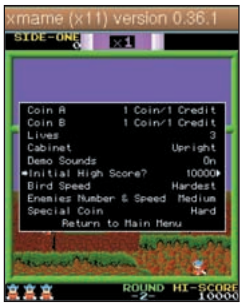
After saving the makefile, type **make** and fix a snack. My Debian 2.2 Linux took just over 40 minutes to compile the default X11 version with recommended optimisations on a K6/2-500. It munched through more than 700 C source files, without a single warning message from *GCC*.

Installation

make install copies the compiled code and documentation into the preset directories. Unless you alter those you'll need root superuser privileges to do this, as by default the files go in system directories to which a normal user does not have write access. Copy the supplied xmamerc.dist file from the /doc directory to a new directory ~/.xmame, and edit it to suit

»

Connect up your ride-on lawnmower and an old PC and *voilà*!



```
 Coin A                 1 Coin/1 Credit
 Coin B                 1 Coin/1 Credit
 Lives                                3
 Cabinet                        Upright
 Demo Sounds                         On
◂Initial High Score?             10000▸
 Bird Speed                     Hardest
 Enemies Number & Speed          Medium
 Special Coin                      Hard
        Return to Main Menu
```

This configuration menu lets MAME users access switches normally hidden inside a *BombJack* cabinet.

Atari's vector scan *Asteroids* delivered far higher resolution than traditional raster-based games.

Taito's *Space Invaders* triggered a small change shortage in Japan.

«

yourself. Most of the files for specific games go in subdirectories of /usr, where they can readily be shared.

The ROM files are copyright and their usage is strictly controlled for legal reasons. The canonical site, 'www.mame.dk', gives legal justifications for their archive, including backups, education and emulator programming. There are no ROMs on our cover discs, but plenty online; *LXF19* included tips on buying original ROM sets.

Filenames and letter case must be just right or *XMAME* will not start a set of ROMs. Remember to check the case of the sub-directory name as well as the ROM; prefix the launch command with **strace** to check the exact paths if necessary. Case dependence is a Unix quirk irrelevant to other *MAME* platforms, so renaming is often necessary.

The mame.dk website includes notes on each ROM set, typically covering gameplay tips, variants, bugs and other history. The only essential command line parameter is the ROM set name – though it defaults to the name set in the config file, typically *pacman* – but almost 200 lines of help are available if you specify the **-?** switch as an argument. Many **-list** options give details of the supported game ROMs and the chips they use. The **-listcpu** variant gives a wide chart of CPU popularity by year; with the Z80, 6809 and 68000 predominating a varied list.

MAME may put ROMs and other shared files in /usr/lib/games/xmame or /usr/games/lib/xmame, depending on the package you have installed. You can change the path by editing xmamerc, but the FAQ is misleading for version *0.57* running under X11; the file needs to be called xmame-x11rc, not xmamerc as used for older versions, and the directives it holds should be specifically for that version, not for other types of display – it will ignore ones for, say DGA, warning if it finds any.

Initial Options

The option **-?** calls up hundreds of lines of documentation about *MAME's* other command line options. You can preset these in a configuration file for each user or ROM on your system, or save typing by running a front-end program. Many front-ends for MAME are available; for instance *MAMECAT* uses Trolltech's *Qt* graphics toolkit and adds graphic previews and game search options, whereas *gRustibus* is tailored for the *GNOME* environment. Portable ones like *it-mame* and *tkmame* are written in Tcl/Tk, and *kmamerun* and *gnomane* are alternatives for *KDE* and *GTK* respectively.

However invoked, useful options include **-scale** to resize the screen. Most games run at around TV-style 300 by 200 pixel low resolution, so they're easily lost on a typical X desktop. A scale

MAME games

Know thy enemy in Williams's Defender.

***Terminator 2* lives to die again - he always said he voz going to be back...**

All of the games that we tested were playable. Performance depends on the ROMs and your Linux system; older games are limited by host graphics speed, especially in X, but emulation of the multiple processors in later cabinets puts more strain on your CPU. *MAME* interprets all game code; 68K Linux might run the code for some games directly, but *MAME* opts to emulate them all, for consistency and portability, favouring x86 or RISC platforms.

Pacman

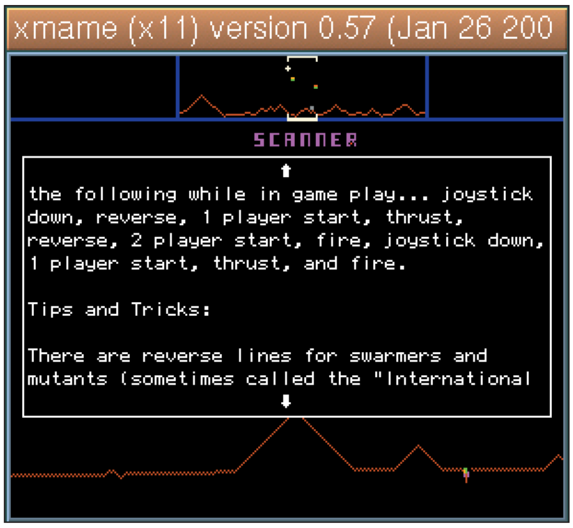
Eighties arcade hits are often playable even on a slowish Linux box. The original *Pacman* ROM set is called *Puckman*, matching the name it was given, and the main character's resemblance to a hockey puck, till the distributors panicked that vandals would find it easy to change the initial letter into an 'F'... Its humble 3 MHZ Z80 delivered less CPU power than a ZX-80, bolstered by Namco sound and 16 colour sprite graphics with 288 by 224 pixels at 60 Hertz, normally orientated vertically rather than horizontally. Sega's *Pengo* was essentially a 32 colour version.

MAME's 'DIP Switches' menu offers an alternative 'Cocktail' tabletop display format as well as *Pacman's* upright default, with Normal or Hard difficulty and an alternate set of ghost names more intelligible to an English audience. Initial number of lives per coin can be set between one and five, with bonus lives available every 10,000, 15,000 or 20,000 points, or none if you're running an arcade where you want to discourage expert players.

Service mode lets you test the key assignments, playing a different sound every time it detects a button-press, joystick move, or coin entry. This was originally intended for engineers checking out the cabinet, but is all part of the authentic *MAME* experience and may be handy if you're building a replica cabinet and want to test controls patched over the computer keyboard.

Eight bit eighties

Defender runs on a 1 MHz 6809 with a slightly slower 6808 driving Sam Decker's awesome DAC effects. After the



the following while in game play... joystick down, reverse, 1 player start, thrust, reverse, 2 player start, fire, joystick down, 1 player start, thrust, and fire.

Tips and Tricks:

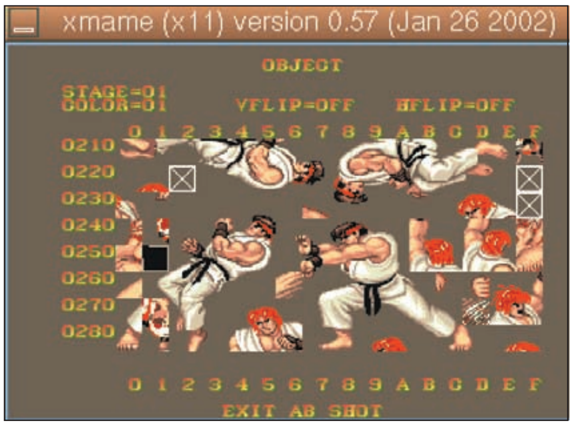
There are reverse lines for swarmers and mutants (sometimes called the "International

***XMAME* overlays tips and trick information onto the *Defender* screen.**

disconcerting startup pattern – while the ROMs are checked – press F2, the internal 'advance' button, to look through the setup options and press 7, mapped to the internal button to clear the high score table, to tweak settings. *Defender's* Game History section, on the Tab menu, has enlightening details of bugs, special tactics, and an 'easter egg' triggered if you do just the right sequence of pressing the five buttons plus joystick moves.

Taito's *Qix* runs on dual 6809s with another Motorola chip, the 6802, delivering sound by direct writes to a 'Digital to Analogue Conversion' DAC much like Linux's '/dev/dsp'. *Qix* features few colours and no scrolling, but *MAME* still has to emulate three synchronised processors to run this exceptionally original game.

The 1980-vintage Atari *Centipede* arcade game runs on a 1.5 MHz 6502 with an Atari Pokey chip, which MAME renders


xmame (x11) version 0.57 (Jan 26 2002)
OBJECT
STAGE=01
COLOR=01
VFLIP=OFF
HFLIP=OFF
0 1 2 3 4 5 6 7 8 9 A B C D E F
0210
0220
0230
0240
0250
0260
0270
0280
0 1 2 3 4 5 6 7 8 9 A B C D E F
EXIT AB SHOT

***Street Fighter 2* has a built-in editor for the sprite graphics.**

factor more than two or three slows X down noticeably, and the chunky pixels become obvious, but you can work around this by using a more appropriate screen mode or a custom screen.

My 500MHz K6 ran *Afterburner 2* on a 16 bit *XFree86 3.3.5* desktop at a playable 20 frames per second, four times the original size via **-scale 2**. To get a full-screen display at the original 60 Hertz I'd need to bypass X. F11 toggles a frame rate counter at the top of the screen, and F8 and F9 adjust the frame skip, trading CPU time for update speed. **-effect 1** uses smoothing as well as double width and height, for still frames even better than the original machine, but updated at a slightly flickery 12 frames per second.

Other **-effect** options mimic the original screen scanning grid or smooth out the zoomed display. Special options for vector displays offer adjustable simulated beam size, flicker and anti-aliasing. **-ror** and **-rol** rotate the display 90 degrees each way, giving accurate portrait-mode displays as many arcade machine screens appear taller than they are wide. **-flipx** and **-flipy** do the same for mirrored displays – less useful unless you play on an autocue machine, or dangling from the rafters.

-samplefreq sets the audio output rate, making a similar trade off for samples, with muffled sound but less CPU load at the low end of this range; use 48000 or configure this for the best quality your machine can deliver. The default is half CD rate, 22050 Hertz. You can add buffers, incurring some lag but smoothing out glitches, with **-bufsize** which sets the number of buffers rather than their size. Some games insist on sound emulation, as they use it for synchronisation, but **-fakesound** allows those to run silently. **-volume** sets audio attenuation, from **-32** up to **0**, the loudest.

MAME delivers *Pacman* - or *Puckman* - pixel perfect.

rather erratically unless you finely tune the sound settings. *BombJack* runs on two Z80s (one for sound) and uses a AY8910 sound chip – like a ZX Spectrum 128.

3D vectors

Atari and Cinematronics vector games presaged modern 3D gaming with analogue wireframe hardware. Atari's rock-breaking *Asteroids* ran on a smooth vector scan display with an 8 bit CPU feeding it co-ordinates at a resolution of 1024 by 768 pixels which raster games could not match, though the refresh rate depended on the number of vectors – flickering as the screen filled up. *Asteroids* used the same physics and control system as *Space War*, developed at MIT in 1961, which begat Atari founder Nolan Bushnell's *Computer Space* a decade later.

Tempest, originally conceived as 'first person *Space Invaders*', tweaked Atari's vector hardware to work in colour, but overheating problems mean emulation is your best chance of a playable game, these days. It ran on a 6502 at 1.5 MHz with two Pokey chips, as used in the Atari 400, giving clicky MAME sound, as for *Centipede*. Mouse emulation is a nice idea but the original used a trackball and direction reversals take some getting used to, but the keys work pretty well, too.

Graphic progress

Konami's *Vulcan Venture* side-scrolling shooter uses 1024 colours and almost 3MB of ROM, including sampled speech. This is a good one to run full screen as it used a landscape display rather than the portrait layout common in arcades but not computer screens other than Radius Pivots.

Capcom's *Street Fighter II* uses a 12 MHz 68000 with a 4 MHz Z80 driving OKI and Yamaha sound chips. The screen resolution is a wide-screen 384 by 224 pixels. It starts up with extensive self-test facilities, including a screen just to try out the sound effects.

The back tick character, top left on many PC keyboards, calls *MAME*-specific options, trimming the overall volume and relative levels of stereo from the YM2151 synth and the companion OKI6295 sample player's four voices. Key and controller signals get sent to the game menu as well as the overlaid *MAME* one, so the arrow keys affect both displays and settings. Work around this by setting *MAME* ones first, then press <Esc> to concentrate on the game ROM configuration later. These *MAME* options also control image brightness and gamma, the ratios between preset and rendered brightness levels. You'd need a screwdriver or soldering iron to adjust these in a real arcade cabinet.

Terminator 2 runs to more than 7MB of ROM, with a 32 bit Texas TMS34010 RISC CPU plus a 2 MHz 6809 for audio, driving a DAC and two more audio chips. The 400x256 pixel display uses a 12 bit colour palette. Hidden features of this game reward exploration when you don't have to keep feeding it with coins. It plays chilling samples during its setup mode, and its attract mode sounds and graphics are quite exceptional.

Hidden menus built into a *Street Fighter 2* cabinet allow graphics editing using the underlying blocks, with vertical and horizontal flip and palette controls, configuration of the text messages, and enough other goodies to virtually constitute a second game, allowing you to make your own *Street Fighter 2 SE*, if not *Street Fighter 3*, by patient experiment. *MAME* has exceptional depth; if you like emulation it merits investigation, even if you've never set foot in an arcade. LXF

MAME Links

The many flavours of Mame

All formats *MAME*:	http://www.mame.net
Arcade ROMs:	http://www.mame.dk
Grustibus front end:	http://grustibus.sourceforge.net
Kemulator front end:	http://kemulator.sourceforge.net
TKMame front end:	http://tkmame.retrogames.com
XMame home:	http://x.mame.net
XMame patches:	http://www.win.tue.nl/~stijn/xmame

***Qix* is one of the most original abstract arcade games.**