

doc/mame

COLLABORATORS

	<i>TITLE :</i> doc/mame		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 8, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	doc/mame	1
1.1	mame.guide	1
1.2	mame.guide/Configuration	1
1.3	mame.guide/UseDefaults	3
1.4	mame.guide/ScreenType	3
1.5	mame.guide/ScreenMode	3
1.6	mame.guide/DirectMode	4
1.7	mame.guide/DirtyLines	4
1.8	mame.guide/Depth	4
1.9	mame.guide/Allow16Bit	4
1.10	mame.guide/FlipX	5
1.11	mame.guide/FlipY	5
1.12	mame.guide/Antialiasing	5
1.13	mame.guide/Translucency	5
1.14	mame.guide/BeamWidth	5
1.15	mame.guide/VectorFlicker	6
1.16	mame.guide/AutoFrameSkip	6
1.17	mame.guide/FrameSkip	6
1.18	mame.guide/Width	6
1.19	mame.guide/Height	6
1.20	mame.guide/Buffering	7
1.21	mame.guide/Rotation	7
1.22	mame.guide/Sound	7
1.23	mame.guide/AudioChannel0	8
1.24	mame.guide/AudioChannel1	8
1.25	mame.guide/AudioChannel2	8
1.26	mame.guide/AudioChannel3	8
1.27	mame.guide/MinFreeChip	8
1.28	mame.guide/Joy1Type	9
1.29	mame.guide/Joy1ButtonBTime	9

1.30	mame.guide/Joy1AutoFireRate	9
1.31	mame.guide/Joy2Type	9
1.32	mame.guide/Joy2ButtonBTime	10
1.33	mame.guide/Joy2AutoFireRate	10
1.34	mame.guide/RomPath	10
1.35	mame.guide/SamplePath	10
1.36	mame.guide/AsyncPPC	11
1.37	mame.guide/Paths	11
1.38	mame.guide/Keys	12
1.39	mame.guide/Tips	13
1.40	mame.guide/Rebuilding	13
1.41	mame.guide/Thanks	14
1.42	mame.guide/Contact	14

Chapter 1

doc/mame

1.1 mame.guide

MAME - Multiple Arcade Machine Emulator

MAME or Multiple Arcade Machine Emulator is as the name suggest an emulator that let's you play hundreds of classical arcade games on your computer. The MAME project was started by Nicola Salmoria and is now maintained by Mirko Buffoni, but there are many contributors from both the PC and Mac world. This is the Amiga port of MAME which is based upon the 0.35 beta 12 PC source.

Configuration

Paths

Keys

Tips

Rebuilding

Thanks

Contact

1.2 mame.guide/Configuration

Configuration

MAME uses a config file called mame.cfg as a configuration database. This file is read on startup and automatically saved on exit. With this database it's possible to have an unqiue configuration for every driver.

To make life a bit simpler it's possible to make a default configuration for the two major types of drivers (bitmap and vector). The configuration can be set from both CLI and the GUI. The CLI configuration options are listed below. The options must be given on the form <option>=<value>.

UseDefaults

ScreenType

ScreenMode

DirectMode

DirtyLines

Depth

Allow16Bit

FlipX

FlipY

Antialiasing

Translucency

BeamWidth

VectorFlicker

AutoFrameSkip

FrameSkip

Width

Height

Buffering

Rotation

Sound

AudioChannel0

AudioChannel1

AudioChannel2

AudioChannel3

MinFreeChip

Joy1Type
Joy1ButtonBTime
Joy1AutoFireRate
Joy2Type
Joy2ButtonBTime
Joy2AutoFireRate
RomPath
SamplePath
AsyncPPC

1.3 mame.guide/UseDefaults

Yes

The default configuration for this type of driver (bitmap or vector) is used instead of the drivers own configuration. (Default)

No

The drivers very own configuration is used.

1.4 mame.guide/ScreenType

Best

The screenmode that looks best suited for the selected driver is used. (Default)

WB

A window on the Workbench is used.

Custom

The screenmode described by ScreenMode and Depth is used.

UserSelect

A screenmode requester is presented at startup where the wanted screenmode can be selected.

1.5 mame.guide/ScreenMode

This option takes a valid screenmode id number as the value.

1.6 mame.guide/DirectMode

Off

Direct access to graphics card memory isn't used.

Draw

The graphics card memory is used directly as MAMEs internal screen buffer. There's no second copy stage, but several layers with transparency may be written to it.

Copy

The internal screen buffer is copied directly into the graphics card memory. In the PPC version of MAME this will be done by the PPC.

Note that both Draw and Copy method requires a graphics card with an 8 bit screenmode and that it can be a bit dangerous because it locks the bitmap during the execution of a driver.

1.7 mame.guide/DirtyLines

Yes

If it's supported by the driver then only changed lines will be copied to the display. (Default)

No

The whole internal screen buffer is copied to the display for every frame.

1.8 mame.guide/Depth

The number of bits per pixel.

1.9 mame.guide/Allow16Bit

Yes

If the driver supports 15/16 bit colors then this option allows MAME to run the driver with 15/16 bit true color instead of the normal palette mode. Note that MAME will fall back to palette mode if the user has selected a screen mode which doesn't support 15/16 bit. (Default)

No

The driver will run in palette mode even if the driver and screen mode supports 15/16 bit.

1.10 mame.guide/FlipX

Yes

Flip the display around the y axis.

No

Don't flip the display around the y axis. (Default)

1.11 mame.guide/FlipY

Yes

Flip the display around the x axis.

No

Don't flip the display around the x axis. (Default)

1.12 mame.guide/Antialiasing

Yes

Draw the vector graphics using antialiasing.

No

Don't draw the vector graphics using antialiasing. (Default)

1.13 mame.guide/Translucency

Yes

Draw the vector graphics with translucency.

No

Don't draw the vector graphics with translucency. (Default)

1.14 mame.guide/BeamWidth

The width of the vector beam. It's an integer from 1 to 16 where 1 is default.

1.15 mame.guide/VectorFlicker

Emulate the flicker on vector graphics based drivers. It's an integer from 0 to 100 where 0 is no flicker and default.

1.16 mame.guide/AutoFrameSkip

Yes

The frameskip value will be dynamically calculated in an attempt to get as close to full speed as possible.

No

The frameskip value is selected by the user. (Default)

1.17 mame.guide/FrameSkip

Number of frames to skip. A maximum of 3 frames can be skipped.

1.18 mame.guide/Width

Width of the display. It only overrides the requested width from the driver if the requested width is smaller. Vector graphics based drivers will scale to this width.

1.19 mame.guide/Height

Height of the display. It only overrides the requested height from the driver if the requested height is smaller. Vector graphics based drivers will scale to this height.

1.20 mame.guide/Buffering

Single

The internal screen buffer is drawn into the display buffer that's currently displayed. This may lead to flickering (Especially on AGA). (Default)

Double

Two display buffers are used so the internal screen buffer can be drawn in the display buffer that currently isn't displayed.

Triple

Uses three buffers to reduce the chances for delay while waiting for a buffer that isn't currently displayed or on it's way to be displayed.

1.21 mame.guide/Rotation

No

Don't rotate the display. (Default)

Left

Rotate the display to the left.

Right

Rotate the display to the right.

1.22 mame.guide/Sound

No

Don't make a sound. This normally speeds up the emulation because most arcade games had dedicated processors for the sound output that in this case don't need to be emulated.

Paula

Uses audio.device to play the sound. In this case only 4 of the 16 channels used internally will be played. Which 4 channels played can be configured with

```
AudioChannel0
,
AudioChannel1
,
AudioChannel2
and
AudioChannel3
.
```

AHI

All 16 channels are played using the lowlevel interface of AHI.

1.23 mame.guide/AudioChannel0

A number between 0 and 15 specifies which of the 16 internal sound channels that will be played in hardware channel 0.

1.24 mame.guide/AudioChannel1

A number between 0 and 15 specifies which of the 16 internal sound channels that will be played in hardware channel 1.

1.25 mame.guide/AudioChannel2

A number between 0 and 15 specifies which of the 16 internal sound channels that will be played in hardware channel 2.

1.26 mame.guide/AudioChannel3

A number between 0 and 15 specifies which of the 16 internal sound channels that will be played in hardware channel 3.

1.27 mame.guide/MinFreeChip

If `audio.device` is used for the sound playing then samples used by the selected driver will be loaded into chipram. Many drivers use many and large samples and using all of the chipram can be rather dangerous. As a solution this option lets you specify how many kilobytes of chipram must be available after a sample has been loaded. If this is not the case the sample will be loaded into fastram instead. It can still be played, but it will be a little bit slower. This option defaults to 64.

1.28 mame.guide/Joy1Type

No

Disable joystick 1.

JoyStickPort2

A joystick connected to port 2 is used as joystick 1.

JoyPadPort2

A CD32 compatible joypad connected to port 2 is used as joystick 1.

MousePort1

A mouse connected to port 1 is used as a trackball instead of joystick 1.

1.29 mame.guide/Joy1ButtonBTime

Most arcade games use at least two fire buttons but a standard joystick has only one. Because of this a second fire button can be emulated by keeping the fire button pressed for a certain amount of time. This options specifies how many 1/10 of a second the fire button must be kept pressed before a second fire button is triggered. The value can be from 0 to 9 where 0 is default and means that a second fire button isn't emulated. This option is mutually exclusive with

Joy1AutoFireRate

.

1.30 mame.guide/Joy1AutoFireRate

Keeping the fire button pressed can be converted into multiple fire button presses. The option specifies the number of times per second the fire button should be pressed. The value can be from 0 to 5 where 0 is default and disables the autofire support. This option is mutually exclusive with

Joy1ButtonBTime

.

1.31 mame.guide/Joy2Type

No

Disable joystick 2.

JoyStickPort1

A joystick connected to port 1 is used as joystick 2.

JoyPadPort1

A CD32 compatible joypad connected to port 1 is used as joystick 2.

1.32 mame.guide/Joy2ButtonBTime

Most arcade games use at least two fire buttons but a standard joystick has only one. Because of this a second fire button can be emulated by keeping the fire button pressed for a certain amount of time. This options specifies how many 1/10 of a second the fire button must be kept pressed before a second fire button is triggered. The value can be from 0 to 9 where 0 is default and means that a second fire button isn't emulated. This option is mutually exclusive with

Joy2AutoFireRate

.

1.33 mame.guide/Joy2AutoFireRate

Keeping the fire button pressed can be converted into multiple fire button presses. The option specifies the number of times per second the fire button should be pressed. The value can be from 0 to 5 where 0 is default and disables the autofire support. This option is mutually exclusive with

Joy2ButtonBTime

.

1.34 mame.guide/RomPath

The selected driver will look for roms in this path as well as in the normal ones.

1.35 mame.guide/SamplePath

The selected driver will look for samples in this path as well as in the normal ones.

1.36 mame.guide/AsyncPPC

Yes

The PPC runs asynchronously with the m68k. In this case two screen buffers are used. One is used by the m68k to update the screen and one is drawn into by the PPC. For every frame the buffers are exchanged.

No

The PPC runs synchronously with the m68k. In this case only one screen buffer is used. This makes games that depend on what they drew previous frame work better. So use this if you see artifacts on screen in games like Space Invaders.

1.37 mame.guide/Paths

Paths

MAME searches for roms in these directories/archives:

- <RomPath>/<GAME>/
- <RomPath>/<GAME>.zip
- <RomPath>/<GAME>.lha
- <RomPath>/<GAME>.lzx
- PROGDIR:roms/<GAME>/
- PROGDIR:roms/<GAME>.zip
- PROGDIR:roms/<GAME>.lha
- PROGDIR:roms/<GAME>.lzx
- PROGDIR:<GAME>/
- PROGDIR:<GAME>.zip
- PROGDIR:<GAME>.lha
- PROGDIR:<GAME>.lzx

MAME searches for samples in the same places it searches for roms and in these directories/archives:

- <SamplePath>/<GAME>/
 - <SamplePath>/<GAME>.zip
 - <SamplePath>/<GAME>.lha
-

- <SamplePath>/<GAME>.lzx
- PROGDIR:samples/<GAME>/
- PROGDIR:samples/<GAME>.zip
- PROGDIR:samples/<GAME>.lha
- PROGDIR:samples/<GAME>.lzx

MAME has internal support for zip files but need external programs for lha and lzx support. If you want to use lha or lzx ROM archives you need to have lha and/or lzx in your default path.

Highscores are stored in PROGDIR:hi/ and game configurations (dip settings, keyboard and joystick setup) are stored in PROGDIR:cfg/.

1.38 mame.guide/Keys

Keys

Right-Amiga-N
New game.

Right-Amiga-Q
Quit MAME.

Right-Amiga-S
Snapshot the display and save it as an IFF ILBM. This doesn't work if
DirectMode
is set to On.

Tab
Enter configuration menu. Press Tab or Esc to get back to the emulation.

,

Show volume, brightness and gamma correction control. Only volume control has any meaning to the Amiga version. Use left and right cursor keys to decrease and increase selected control. Use up and down cursor keys to move between different controls.

P
Pause game.

F3
Reset game.

F4
Show the game graphics. Use cursor keys to change set/color, F4 or Esc to return to the emulation.

F7

Toggle speed display.

F8

Change frameskip on the fly.

F10

Toggle speed throttling.

1.39 mame.guide/Tips

Tips

1. Every NeoGeo driver needs neo-geo.rom which is the NeoGeo BIOS. The best thing to do is to put this file in the directory neogeo or roms/neogeo so it can be accessed by all NeoGeo drivers.

1.40 mame.guide/Rebuilding

Rebuilding

If you want to rebuild MAME yourself you need both the Amiga source and the PC source it's based on. The latest Amiga source is available at [`http://www.triumph.no/mame.html`](http://www.triumph.no/mame.html) and the PC source should be available at either the official homepage or [`http://www.davesclassics.com/mamepage.html`](http://www.davesclassics.com/mamepage.html).

Software needed:

1. A GeekGadgets installation. For more information check out [`http://www.ninemoons.com`](http://www.ninemoons.com).
2. Devpac or a source compatible assembler. You'll have to replace the genam command in the makefiles if you don't use Devpac.
3. If you change src/amiga/mame.cd you'll need CatComp or a compatible clone.

Building steps:

1. Unarchive the Amiga source somewhere.
 2. Make a directory called org/mame where you installed the Amiga source.
-

3. Run `make -f makefile.amiga mamefix`. This will modify all the PC sources so they're compilable with gcc.
4. Run `make -f makefile.amiga makedir` to make all the directories needed.
5. Run `make -f makefile.amiga all` to make all versions.

Building under APUS:

1. Make standard Amiga includes accessible under `/include`.
2. Do building steps 1-4 under AmigaOS.
3. Run `make -f makefile.mameppc mameppc.elf APUS=1` under APUS. This will build `mameppc.elf` but the `mameppc` front end and all the m68k versions must still be built under AmigaOS.

Notes:

1. You'll probably get an internal compiler error on `src/amiga/video.c`. To get around this you should compile it with `-O0`.

1.41 mame.guide/Thanks

Thanks

Trond Werner Hansen <trond.hansen@triumph.no>
For his CGXHooks linker library.

Mikael Kalms
For his chunky to planar routine.

Børge Nøst
For testing the 68060 version and for not getting more than slightly mad everytime it crashed his machine.

Thom Heppleston
For the gift. :)

1.42 mame.guide/Contact

Contact

Email

<mats.hansen@triumph.no>

Snail

Mats Eirik Hansen
Kirkegårdsveien 13
3616 Kongsberg
Norway

IRC

meh in #amiga on ANet.

Homepage

`'http://www.triumph.no/mame'`

The official MAME project homepage

`'http://mame.retrogames.com'`
