LeoUtils

Henrik Herranen and leopold@cs.tut.fi

COLLABORATORS

	TITLE : LeoUtils	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Henrik Herranen and leopold@cs.tut.fi	August 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

LeoUtils

1.1 Mandela - Table of contents

|\/| _ ._ | _ | _

| | ’_| | | .-| |_) | ’_|

| | (_| | | |_| |_, |_ (_| 1.3

_

___ ___ ’_) _

/ __ / (__ . / \

The FAST / __ / -|- (resolver!

/__ /__ ` _/

Please note! There are two versions of Mandela, the registered and the unregistered version. This document will cover only the registered version.

1.2 Disclaimer

Disclaimer

This product has been extensively beta-tested among people with a great variety of Amigas from a basic A500 w/ 1 Meg memory to the wonderful A4000. I've been running this piece of software through Enforcer and have great confidence in that Mandela runs properly in almost all conditions. At least I have not been able to crash a published version after version 0.991.

Still: The author (me) cannot be held liable for the suitability or accuracy of this software and/or this manual. Any damage directly or indirectly caused by the use of this software is the sole responsibility of the user.

In plain English: In the very unlikely case that your HD crashes, your monitor blows up and your girlfriend leaves you while using Mandela and you think Mandela is the cause of it, I'll in no case get you new ones. However, if something is wrong with Mandela, feel free to send me bug reports.

1.3 Copy right, not wrong!

Copyright

Mandela is not in the public domain. It is Shareware, and the demo version is freely distributable. The program must always be distributed with this document file, in its unmodified form.

There are two versions of Mandela, the unregistered and the registered version. All functions are not available in the unregistered version. The unavailable menus are marked in parenthesis. The source code is provided with the registered version.

THE REGISTERED VERSION OF THIS PROGRAM MAY NOT BE PUT ON ANY PUBLIC DOMAIN LIBRARIES! IT MAY BE COPIED ONLY FOR BACKUP PURPOSES! If I ever find out of a registered version of Mandela that is used by someone else than the people in this list, the developement of Mandela will cease.

The UNREGISTERED VERSION of this program may only be distributed for the cost of the media.

If you have any suggestions or remarks about Mandela, please write to me. Bug reports are especially welcome.

How to register Mandela?

1.4 Why and how to register Mandela

Why and how to register Mandela

Why:

1. The unregistered version of Mandela has some features disabled.

2. You'll get a diskette with the following goodies:

- The latest registered version of Mandela with all menus of the

unregistered version in parenthesis activated.

- The source code of Mandela

- A useful program package called LeoUtils with source code (some

Utils can also be compiled under Unix):

- MemTest: A simple memory tester

- GuideCat: Type AmigaGuide documents on Shell or VT100 terminal

(ANSI-C)

- FakeCPU: Lie about your computer's CPU and/or FPU to the OS

- Words: Counts different words (ANSI-C)

- LeoTune: A very intelligent fortune program (ANSI-C)

- LeoLib: A replacement for ctype.h with ANSI-C character set

(ANSI-C)

- LeoTime: Takes time on program execution (like Unix's "time")

- LeoClock: The most font-sensitive clock program ever made in one

disc block!

- Leopold.font and LeoProp.font: Very clear bitmap fonts for use

as system fonts (and Amiga system with square pixels).

3. You tell me to go on with software developement.

How:

1. Fill in the registration form as described in the file Registration.

2. Send me the registration form with US$20 or 100 FIM in a letter.

3. Wait for my reply!

1.5 System requirements

System requirements

Short version

Amiga model: Any

Operating system: 2.1 required, 3.x adds functionality

Processor: Any, math-coprocessor recommended

Free memory: 1 MB recommended

Long version

Mandela requires that you are running at least OS 2.1. However, if you are running a newer version of Amiga OS, you will get added functionality.

If you have OS 3.0, you can do color cycling and some other more subtle features are enabled, like 24-bit color palettes and sleep pointers.

For saving and loading images, colormaps and Mandelbrot coordinates Mandela uses the public domain iff.library by Christian A. Weber. This library is included with Mandela. If you previously haven't had the library, or you have an older version (this is 23.2), please copy this version to your Libs: directory.

Mandela will run with any processor / math coprocessor setup, although it can operate much more efficiently with faster processors and especially if the system is equipped with a FPU.

Mandela works just fine with GigaMem, although the contour crawling algorithm Mandela uses may sometimes cause lots of disk swapping.

1.6 Introduction

Introduction

So, why again and again new Mandelbrot programs? Here's some reasons:

+1) Speed. Of course, this is said in every Mandelbrot program document file, but I really think this is one of the very fastest Mandelbrot programs for the Amiga ever made. Mandela is written in C with the exceptions of some small time-critical parts, which are in assembler. Special automatic 68020 and 68881/2 code is used in the assembler code.

+2) Easiness. I think Mandela's user interface is fairly friendly. For instance, you never have to wait that the calculation finishes (or explicitly stop it) to get control over other functions.

+3) Size. Mandela is one of the smallest Mandelbrot programs at the moment. This is partially because Mandela lacks some features present in some other Mandelbrot programs, and partially because I've put great effort in keeping this piece of software as compact as possible.

+4) Internal multitasking. Mandela can go on with calculation even when you are editing colors.

+5) Intelligent priority control. While the user interface may be set on a high priority the calculation process may be set to another priority so that it won't affect other tasks that want processor time.

+6) Can load coordinates from pictures created with (virtually) every Amiga Mandelbrot program. Can read and write both MAND and MANC chunks.

+7) No Enforcer hits, no mysterious hangups, no computer crashes. Every effort has been made to keep bugs out of Mandela.

+8) Mandela can use HAM on older Amigas to emulate a 256 color mode.

+9) Mandela can nicely scroll colors under OS 3.

+10) Locale support

And, for contrast, some minuses of Mandela:

-1) Animations not supported. May be some day.

-2) No ARexx interface. Too bad. Maybe I'll learn some day how to use ARexx.

-3) The unregistered version of Mandela is crippled.

1.7 Author - Leopold-Soft

Leopold-Soft

Wonderful (?) software since 1982.

My address is:

Henrik "Leopold" Herranen

Opiskelijankatu 38 A 7

33720 Tampere

Finland

Internet e-mail: leopold@cs.tut.fi

How to register Mandela?

1.8 The history of Mandela

The history of Mandela

Explanations:

The letter 'b' attached to a version number means the version was sent

to Beta-testers.

The character '*' attached to a version number means the version has

been published on the Internet.

Mandela history:

1.3 * 220K, 61K

970127 Improved dithering. Made Threshold use dithering if possible.
970125 Added dithering. Optimized display and contour crawling speed.
1.23 * 215K, 60K
960323 Added localization support and three languages: English,
Suomi (Finnish), Deutsch.
1.21 * 201K, 57K
950418 Allowed using of smaller than 320 x 200 calculating areas.
However, the minumum size of the screen is still always
320 x 200 pixels. Too bad, time functions made the code
grow a bit.
Saving pictures is now disabled in the unregistered version.
1.2 * 201K, 55K
950317 By accident Mandela had become OS 3 only. Made functions to
emulate OS39's Alloc/FreeBitMap() with OS functions available
to OS 2.
950316 Move()s and Draw()s taken away even when Continuous update is
selected (first draw the outlines, and only after the whole
scanline update it.
1.12 b 198K, 55K
950315 Mandela now can save either the whole screen as it is seen on
the display or only the raw Mandelbrot image. The added layer
of abstraction slowed saving down quite a bit, but it still is
fast enough.
950314 A major rewrite has been done to the display handling. All
WritePixel()s, Move()s, Draw()s and SetAPen()s are gone unless
you have the Continous update option set on. However, the
graphics code still is totally OS friendly, thanks to
WritePixelLine8(). The graphics code got smaller and a lot
faster. This difference can especially be seen with HAM
display modes.
1.11 199K, 55K
950311 Mandela has learned how to save images for other than standard
Amiga screen modes, so now it can save screens of different
display cards.
950310 A bug in saving coordinates corrected: Mandela wouldn't always
save the whole 252-color colormap.
1.1 197K, 54K
941121 Mandela learned how to save MAND chunks with images. Now
Mandela's images can be loaded virtually everywhere.

941116 Mandela learned how to save MANC chunks with images.
1.021 * 193K, 53K
941115 After releasing Mandela’s registered version found out that
starting color scrolling with 2 colors crashed everything.
Inactivated scroll menus if <= 4 colors.
1.02 b 193K, 53K
941115 Mandela uses now iff.library to load/save IFF ILBM. The code
got smaller and bitmap images can be saved (although without
MAND or MANC chunks).
Bugs emerge as you program: on OS 2.x Mandela would try to
stop the color scrolling interrupt although it didn’t exist.
Result: Crash on exit on pre-3.0 Amigas.
1.01 b 198K, 55K
941113 Added scripts and programs to update user versions.
941112 Removed a nasty IDCMP bug I got rid of in version 0.85 but
somehow managed to rewrite.
Added color cycling for OS3 users.
1.0 184K, 54K
941106 A dramatic change in looks. Window became a backdrop window.
The title bar of the screen can be turned on and off. A
special information window replaces the window border
information. If using OS 3.x, the user have the menus
attached also to this little window. A grid is drawn before
the calculation begins to let the user see that something is
really happening when he autoscrolls.
0.994 181K, 53564
941105 A somewhat crippled unregistered version created with
conditional code compilation.
941102 Code cleanup.
A minor bug fixed. Caused sometimes random pixels to the left
border of the Mandelbrot image.
A "Don’t draw as you search borders" option added.
0.993 b 181K, 53544
941031 Rewritten calculation table routines in assembler. with
disabled error checking. Of course the code got somewhat
faster and 4K smaller (I used to inline the table functions).
The original C version with error checking not deleted (better
to be in place when developing Mandela further).
0.992 179K, 57648

941029 Again fixed the very bad bug (see v. 0.991b). Hope it works
this time.
Added some more verbosity to the user interface: Window border
messages for new precision and MaxIter.
0.991 b 174K, 57188
941024 Fixed a very bad bug in launching of the calculation process,
which caused random lockups and weird behaviour.
Added HAM, not HAM, DOUBLEPAL and DOUBLENTSC to the list of
rejected modes. Made HAM a default rejected mode if Mandela
finds out there are 8-bit display modes.
0.99 b 173K, 58088
941024 Implemented the coordinates window. Had finally to include
the printf() -function; thus Mandela grew several KBs. Sigh.
But, only Beta-testing left to do :-) . And the documents :-/
0.922 168K, 51344
941024 Halfbrite and dual playfield screen modes disabled from the
Screen mode requester. Also other filters were made.
0.921 164K, 50316
941024 "Beep on end of calculation" -mode selection added. Minor bugs
fixed. Commant: Mandela seems to be very well configurable
nowadays. To do: Must enable the "Coordinates" menu selection.
941022 Automatic calculation precision provided. Preferences toggle
menu "Always load colors" created. "Save Coords" implemented
and corresponding menu enabled (MANC chunk saved, but MAND not
until I get my hands on the Motorola 96-bit floating point
format book).
0.92 158K, 48648
941022 The menu system is totally rewritten to the way it is
recommended to do (using GadTools et all).
Some minor fixes: Font in Edit Colors works better under OS 2,
Priority control enhanced (User Interface priority is now
selectable), "Do you really want to do this and that"
requesters can now be disabled or enabled at wish.
Thus totally more than 20 KB of code was rewritten for this
version.
0.911 158K, 46880
941020 The 32-bit 68000 calculation code rewritten in assembler
causing the calculation speed to more than double. Now all
single point calculation programs are in assembler.

0.91 157K, 47684
941019 Major change: Mandela became multithreaded! The calculation is
now a Process, so it is possible to, say, edit colors while
the image calculation is going on (with the exception of HAM
mode)! The user interface is now also much nicer because it's
always running at high priority. No more lags on a heavily
loaded Amiga.
A 20-step history of the last zoomed positions is maintained.
If you zoom too deep and then want to back out a bit, just
press Amiga-Z.
0.901 151K, 46352
941018 Assembly routine made for floating point calculation using the
Amiga OS mathieeedoubbas.library. Didn't work faster than the
C language version, though :-(, because all the time is spent
in the s_l_o_w OS library calls.
Made a new 16-bit integer routine for the 68020 processor and
modified the 32-bit 020-processor routine to take advantage of
instruction pipelining: two consecutive commands rarely touch
the same registers and commands have been inserted between
consecutive multiplication commands.
941016 Minor modification to MMisc.c. Perhaps several thousand clock
cycles saved per picture (the calculation functions are now
accessed through a pointer rather than a complex switch
statement).
0.90 152K, 43712 bytes
931026 Added a boost priority preference, which allows the program to
raise its priority every time it is used interactively.
The system requesters are redirected to Mandela's own screen.
931025 Corrected a bug which made it impossible to open the Edit
Colors menu under 2.x. Actually I think it was not a bug in my
program, rather weirdness of OS 2 :-)
Added version string, which makes it possible for a user or
script to say "Version Mandela".
Made preferences to load/save from/to Mandela's own dir
(PROGDIR:Mandela.prefs) instead of the current dir. No more
Mandela.prefs files all around your HD.
Color load/save requester defaults to PROGDIR:Colors and ILBM
load/save requester defaults to PROGDIR:Pictures. Maybe this
default directory thing should be configurable?

931021 Loading and saving colormaps has become a reality.
0.883 b 137K, 39160 bytes
931018 Got rid of the autogenerated gadget code and finally
implemented the Edit Colors menu. I really like it. It's fast,
even with HAM. The sinus editor is not implemented, yet.
930810 Info now shows the actual compilation date of the program.
930809 Replaced SetRGB4 to SetRGB32 if running under OS 3. Internally
still using only an 8-bit per color-component palette. Anyhow,
should perform very well on AGA Amigas.
930714 Corrected a bug in Threshold: in HAM mode the drawing
color would be wrong for the calculation.
930531 Added Weird 32-calculation.
0.882 b 156K, 36968 bytes
930512 Created a new 68881-optimized Double Float routine in
assembler gaining a 4-fold (!) increase of speed. The routine
should take advantage of 68882 command pipelining and thus
making it a bit faster, but being a 68881 owner (which doesn't
have this capability) I can't test it.
0.881 153K, 36748 bytes
930511 Reading MANDs is now a reality, thanks to Mika Achren and
the Motorola book he lended me.
Converted the Int16 routine to assembler. Speedup of 25%.
Created a new 68020-optimized Int32 routine in assembler
gaining a 2.5-fold increase of speed. Now Int32 takes only 10%
more time than Int16 on the 68020.
930510 Added Save Preferences menu selection & auto read at startup.
Added Open menu selection. Now I can read MANC chunks from
ILBM files. MAND seems to be a bit weird with its 12 byte
floating point values. I really should be reading to the test
which begins today at 9 am.
0.88
930508 Added some sanity checking: If new screen is of same
dimensions as the old one, don't free calculation buffers.
930506 Ha-haa! By accident I found out how to get the pixel aspect
ratio. I'll drink the Coke myself!
930505 Added low-memory condition checking and handling. It seems to
be somewhat flakey, but I suspect it's the fault of the OS and
not Mandela.
Changed HAM image rendering. Result: HAM fringing in virtual

screens minimized. Other side of coin: drawing (which was
chanched to WritePixels from highly optimized Moves & Draws)
unfortunately slowed down especially under OS 2. Too bad.
930504 Screen mode requester introduced. It sure is a nice thing.
The screen pixel aspect ratio calculated... hmm... uglily...
Tell me how to do it correctly and I'll buy you a Coke.
0.87 89K, 26172 bytes
930429 Deleted the HiRes, SuperHiRes, Interlace and Colors
selections. Going to replace them with the standard OS 2.1
screen mode requester. Gotta get OS 3.0 include files!
930428 Changed an idle loop of Delay(5); loop() to Wait(), thus
Significantly saving processor time :-)
930424 Added 3.0 menu color compatibility. Believe me or not, but I
actually guessed the value of WA_NewLookMenus! Sure would
like to get those damned 3.0 include files!
930423 Made menus 100% font sensitive, with proportional widths et
al. Maybe should replace custom routines with the OS 2.0
standard ones. Sigh.
930421 Added screen flashing when finished if calculation took over 1
minute.
930420 Enhanced the operation of calculation time estimate display to
avoid ridiculous and too much fluctuating values when making
long, multi-hour calculations.
930420 Added priority control.
0.86 b
930419 Fixed calculation time estimate display.
930418 Fixed a bug, which in my opinion was not in Mandela, but in
the OS 2.04 in SetWindowTitles: If I use the form
SetWindowTitles(Win, (UBYTE *) ~0L, "String");
the OS would sometimes freak out with hundreds of Enforcer
hits when closing the window. Very odd.
0.85
930417 Divided source into several files instead of a big one.
930410 Removed a nasty IDCMP bug in 0.84, which could cause Mandela
to try to read from an IDCMP pointer no longer available.
0.84 b
930408 Think I got rid of the Enforcer hits. Not much functionality
added.
0.82 b 44K, 14736 bytes
930401 First version sent to Beta-testers. HAM works somewhat, screen
mode selection has flaws. Lots of Enforcer hits.
930315 First lines of code written.

1.9 The basic operation of Mandela

The basic operation of Mandela

Starting Mandela

You may start Mandela either from the CLI interface or by double-clicking its icon. If you start it from the CLI, notice that the default 4096 byte stack is sufficient for Mandela.

When you start Mandela, it will start two processes: one for the User Interface (UI) and one for the Calculation Process (CP). The priorities of the UI and the CP may be changed. The main idea is, anyhow, that the UI has a higher priority than the CP to give you fast access times to Mandela's functions. Also, the default priority of the CP is lower than the default Amiga priority value 0, so it won't slow down your other interactive processes.

If you start Mandela for the first time, you'll next get a warning message that it can't find a preferences file. Later, after you have set Mandela to your liking, you may save your preferences.

If you start Mandela for the first time, it will initially copy the screen mode of Workbench and use it for its own screen. The only difference is that Mandela uses only 2 colors for its initial screen. Usually this display mode is not what you want, and I recommend to change it. A good mode to begin with for non-AGA Amigas is PAL:Low Res with 32 colors and Text Overscan, and for AGA Amigas some low resolution mode with 256 colors.

The total time estimate

After the initial setup Mandela will begin to calculate its default picture.

If the calculation of an image takes more than 10 seconds, and you have told Mandela to show its info window, Mandela will tell you how long it has been calculating and an estimate of the total time.

The estimate is calculated very crudely with a simple formula: Calculated time * Total lines / Calculated lines (a calculated line means a line that has been fully calculated and drawn). However, usually the estimate is sufficiently accurate.

No estimates are given before 10 first scanlines are calculated to avoid ridiculously bad values. Also only one estimate is given per each calculated line for the same reason. Thus, it may be possible that you don't get a new estimate for quite a long time if calculating a big area that takes lots of time.

Zooming in

You may zoom around in a picture with the mouse or with the various zoom menus.

After zooming in several times you may want to change the screen mode to HAM on non-AGA Amigas to simulate a 256-color display mode. You may also want to add the maximum iterations if you have zoomed very deeply into the zones where there is much black color.

Menus

Most of Mandela's functions can be controlled by the standard Amiga drag-and-drop menus.

1.10 Zooming in with the mouse

Zooming in with the mouse

You may zoom in in a picture by setting the mouse pointer to the upper left corner of the are you want to zoom into, press the left mouse button down, and drag the mouse to the lower right corner of the area.

You don't have to wait for Mandela to finish its calculation before you can zoom in.

1.11 The HAM display mode

The HAM display mode

Usually a computer display mode can have 2^n colors, where n represent the amount of bits used per pixel.

Older Amigas could display only 6 bits per pixel, so they could basically only access 2^6 = 64 colors. Even of them only 32 are really usable because there were only 32 color registers and the last 32 colors in the so-called HalfBrite-mode were only darker copies of the first 32 (that's why Mandela doesn't let you choose HalfBrite modes).

However, the creators of the Amiga invented how they could (with a simple, yet clever technique) code 4096 colors in 6 bits per pixel. This mode is called Hold And Modify (HAM).

The HAM mode is not as good as having 12 bits per pixel (2^12 = 4096), but it's very near that. That's why I taught Mandela how to use HAM so that even my old Amiga 2000 could render images with many colors.

However, I must admit that 256 colors in the newer A1200 and A4000 look a zillion times better than this simulation. That's mainly because they can use 256 colors in all display modes whereas HAM is limited to lo-res. Also, they get to choose their 256 colors from a 16.8 million color palette, whereas HAM has only a 4096 color palette. There is also this little nasty feature of HAM called color fringing which I'm not going to discuss now.

1.12 Menus

Menus

The menus of Mandela are as follows:

Project »

New

Open »

Save »

Info..

Quit

Calculation»

Precision »

Max.Iterations »

Threshold »

Coordinates..

Recalculate

Prefs »

Screen mode..

Edit colors..

User priority »

Calc priority »

OK prompts »

Beep after calc. »

Reject modes »

Color scrolling »

Save mode »

Dithering »

Continuous update ×

Load colors w/ images ×

Show title bar ×

Show info window ×

Info window follows ×

Zoom »

In 2X

In 5X

In 10X

Out 2X

Out 5X

Out 10X

Previous

1.13 Menu Project

Project

Project contains the basic operation you need: loading and saving configurations, information, and, of course, quitting the program.

Project »

New

Open »

Coordinates..

Colors..

Save »

Image..

Coordinates..

Colors..

Prefs

Info..

Quit

1.14 Menu Project/New

New

New allows the user to reset to the default Mandelbrot coordinates.

1.15 Menu Project/Open »

Open »

Open lets you open either the coordinates or the colors made by Mandela or some other programs.

Open »

Coordinates..

Colors..

1.16 Menu Project/Open/Coordinates..

Open coordinates..

Open coordinates loads coordinates from an IFF ILBM file with a MAND or MANC chunk. I don't know of any Amigan Mandelbrot program that can save images and don't do it in one of these formats. If the IFF ILBM file contains a CMAP chunk and you have defined you want to load colors with images, the color palette of the picture will be loaded. Also some options set at the Edit colors window will affect loading of colors.

Mandela can also save coordinates.

1.17 Menu Project/Open/Colors..

Load colors..

Open colors loads the colors from an IFF ILBM file with a CMAP chunk. Virtually all of Amiga's graphical software can create such files.

Some options set at the Edit colors window will affect loading of colors.

Mandela can also save colors, and it does so also if you ask it to save coordinates.

1.18 Menu Project/Save »

Save »

Save lets you save some properties of your projects.

Image..

Coordinates..

Colors..

Prefs

1.19 Menu Project/Save/Image..

Save image

Save image will save the current screen to an IFF ILBM file.

Also special Mandelbrot chunks called MANC and MAND are saved. Thus images saved with Mandela can be loaded to virtually any other Mandelbrot programs on Amiga.

1.20 Menu Project/Save/Coordinates..

Save coordinates

Save coordinates saves the coordinates and colormap of the current Mandelbrot rendering as an IFF ILBM file.

The colors are saved as a 256-color CMAP chunk, and the coordinates are saved to both MANC and MAND chunks. Virtually every Amigan graphical software can read this format. The coordinates can later be reloaded by Mandela or virtually any other Amigan Mandelbrot calculating program.

1.21 Menu Project/Save/Colors..

Save colors

Save colors will save the colors of the current Mandelbrot rendering as an IFF ILBM file.

The colors are saved as a 256-color CMAP chunk. Virtually every Amigan graphical software can read this format. The colors can later be reloaded by Mandela.

1.22 Menu Project/Save/Prefs

Save prefs

Save prefs will save your current Mandela calculation and preferences settings as the default values. Next time you start Mandela, these settings will be in use.

1.23 Menu Project/Info

Info

Info will give you some basic information of Mandela.

1.24 Menu Project/Quit

Quit

Can you ever guess what this is for? If not, tough luck.

1.25 Menu Calculation

Calculation

Calculation gives you control over the Mandelbrot image calculation.

Calculation»

Precision »

Max.Iterations »

Threshold »

Coordinates..

Recalculate

1.26 Menu Calculation/Precision »

Precision »

Precision lets you define one of four calculation precisions depending on your needs. The better the precision, the deeper you can zoom in, and the slower Mandela runs (this is not the case on all Amiga models, see below).

The selections you can make are:

Int 16

This precision uses 16-bit integer fixed point arithmetic to do the Mandelbrot calculations. This is by far the fastest way to do calculation on basic Amiga model with a CPU with a 16-bit multiplication command (68000 and 68010).

Int 32

This precision uses 32-bit integer fixed point arithmetic to do the Mandelbrot calculations. The fully 32-bit processors (68020, 68030 and 68040) can multiply 32-bit integers without problems, but on the older processors every 32-bit multiplication must be emulated with from 2 to 3 16-bit multiplications. Thus, Int 32 is almost as fast than Int 16 on 32-bit processors, but somewhat slower on 16-bit processors.

Float 64

This precision uses IEEE double precision floating point arithmetic to do the Mandelbrot calculations. If a math coprocessor is available (68881 or 68882), this will be a reasonably fast mode; about half the speed of Int 16. The internal FPU in the 68040 makes this mode very fast: about as fast as Int 16. However, if you don't have a math coprocessor, Mandela will use the OS routines to emulate the math coprocessor, and that is r_e_a_l_l_y s_l_o_w. As a matter of fact, so slow that you usually don't want to use this mode at all.

Weird 32

While developing the Int 32 -calculation routines I made a range check programming error. However, I liked the result so here it is as a separate precision selection. The speed it comparable to Int 32.

Auto

If you don't want to mind yourself about setting the right precision, you may ask Mandela to automatically set the right precision for you. It will do so every time you zoom or load an image. Selecting this precision mode will exclude all other precision selections.

1.27 Menu Calculation/Max.Iterations »

Max.Iterations »

Max.Iterations lets you select the maximum number of iterations Mandela may do for one Mandelbrot calculation point. The higher the value is the slower Mandela will run in deep areas and the better the image quality will be.

The selections you can make are:

30 62 126 254 510 1022 2046 4094 8190 16382 32766 65534

If you choose an iteration value larger than 254, you that the whole image will have to be recalculated. If you haven't switched the warning off, you still may cancel your menu selection.

1.28 Menu Calculation/Threshold »

Threshold »

Normally you get one color for each Mandelbrot iteration. However, with very deep objects, you may want to have the color changing only for every other, or fourth iteration count. This may be achieved by selecting one value from the Threshold menu.

If you are running Mandela with a non-HAM screen and without any dithering, Mandela will just show the same colour for n steps. Otherwise, the colors between your actual palette colors will be represented as accurately as possible.

The selections you can make are:

1 2 4 8 16 32 64 128

The screen will be updated immediately to reflect the new threshold value. If the calculation was not finished when you chose this menu selection, it will resume only after the display has been updated.

1.29 Menu Calculation/Coordinates..

Coordinates

Coordinates will let you set numerical coordinates for the Mandelbrot complex area.

You will be presented with a requester with three lines that have the values of the current coordinates and that you can set. The values are:

Minimum Real component, Maximum Real component and Center of Imaginary component. The following rules define the values of the components:

-2.2 <= MIN_REAL < MAX_REAL <= 1.3

-1.4 <= CENT_IMAG <= 1.4

The way of defining these values is somewhat different to most other Mandelbrot programs, where you define lower and upper limits also for the Imaginary component. I have chosen this approach, because Mandela keeps the aspect ratio of the image always automatically correct, and defining all the four limits would break this.

1.30 Menu Calculation/Recalculate

Recalculate

Sometimes you may want to recalculate your already calculated image. This may happen if you, for instance, have changed the precision or maximum iterations.

1.31 Menu Prefs »

Prefs

Prefs lets you set the basic settings you want to use. You may also save your preferences (also all calculation settings will be saved).

Prefs »

Screen mode..

Edit colors..

User priority »

Calc priority »

OK prompts »

Beep after calc. »

Reject modes »

Color scrolling »

Save mode »

Dithering »

Continuous update ×

Load colors w/ images ×

Show title bar ×

Show info window ×

Info window follows ×

1.32 Menu Prefs/Screen mode..

Screen mode..

This selection lets you use the screen mode and display size you want Mandela to use. There are several considerations you should make in selecting your screen mode:

The screen mode requester only works with OS 2.1 and later. If you are running OS 2.0x, you must update your asl.library to v38 to be able to use this feature of Mandela. And, of couse, Mandela will not be very usable without a screen mode selector.

To make quick tests take a screen mode with little resolution and render your final versions in higher resolution.

Mandela can make use of Amiga's HAM6 display mode to emulate 256 colors from a 4096 color palette. This will be by far the best colormode for older Amigas with the old graphic chips. It is somewhat slower to draw in HAM compared to other modes, but the result is usually worth the little wait. On new AGA-equipped Amigas where you can access 256 color modes out of a 16.8 million color palette, it is recommended that you disable HAM modes alltogether.

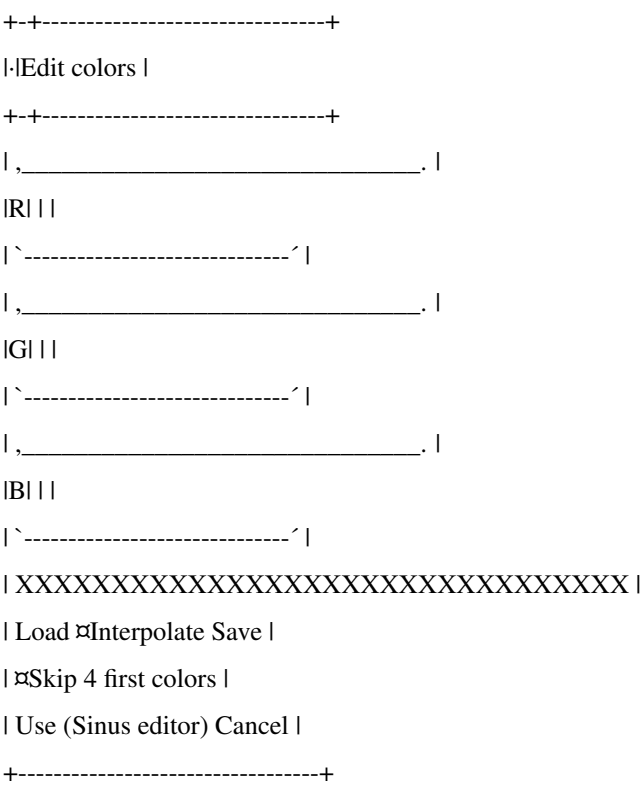
If you do select a HAM mode on an AGA-Amiga, you may select HAM8 (16.8 million colors). This is, however, not a valid selection, and you will be forced to HAM6 (4096) colors by Mandela.

You may select a screen that is larger than your actual display area. This way you may create very big and attractive images. You may scroll the display by moving the mouse to where you want to go.

1.33 Menu Prefs/Edit colors..

Edit colors..

Edit colors will let you edit your default colormap. You will be presented with a window, which's crude Ascii representation is below:



You will see the current 252 color colormap at the line marked with X's. If you have less than 256 colors in your current screen mode, you will get the same approximation of the colormap Mandela uses in the image calculation. You may edit the colors by changing their Red, Green and Blue components at the R, G and B subwindows. The better the vertical resolution of your screen mode is, the more precisely you can edit the colormap. This will usually only have effect in Amigas with the AGA graphic chips.

The options are as follows:

Load is similar to the menu selection Open colors.

Save is similar to the menu selection Save colors.

Interpolate affects the way Load and Save works. Mandela uses a 252 color colormap, and if the colormap to be loaded has less than that amount of colors, Mandela will widen the colormap to fully cover its 252 color colormap. If Interpolate is off and a smaller colormap is loaded, only the first colors are changed.

Skip 4 first colors tells Mandela to skip the 4 first colors when loading a colormap. Many programs, including Mandela, use the first 2 or 4 colours to screen borders, and only the rest of the colors will be actual image data. Loading these 4 first colors will usually result in poorer quality than skipping them. This option is automatically set, if the colormap to be loaded has 256 colors or more.

(Sinus editor) is not usable yet.

1.34 Menu Prefs/User priority »

User priority »

User priority lets you select the priority in which Mandela's user interface is run. You may either select Inherited, or you may select one of the values 5 4 3 2 1 0 -1 -2 -5 -10.

If you use InheritedPri, Mandela will use the same priority that its starter process had (usually 0). The calculation process will get a priority that is 1 less than the UI priority.

If you select a spesific value, Mandela's UI will run at that priority. You may not select a lower value to the UI than you have chosen for the calculation process. This will usually make some priority values inactive.

In Amiga priorities a higher value is better and 0 is the default. My recommendation for the UI priority is 1. That way the UI will be smooth and it will respond quickly to your actions.

1.35 Menu Prefs/Calc priority »

Calc priority »

Calc priority lets you select the priority in which Mandela's calculation process is run. You may select one of the values 4 3 2 1 0 -1 -2 -3 -6 -11.

If you are using the inherited priority for your user interface, the calculation process priority can't be selected.

When you have selected a value, Mandela's calculation process will run at that priority. You may not select a higher value to the calculation process than you have chosen for the UI.

In Amiga priorities a higher value is better and 0 is the default. My recommendation for the calculation process priority is -1. That way it will not affect interactive tasks and while calculating a slow image you can forget the whole existence of Mandela.

1.36 Menu Prefs/OK prompts »

OK prompts »

There are several occasions where Mandela wants to ask the user if it's OK to do something or not, or it just wants to give you a message. From experience I knowthat all the people do not want to see some messages over and over again, while other like the feel of safety they get with them. Thus, the following messages can be turned on and off:

Mouse zoom

If you zoom with the mouse, you will be asked if you really wanted to do that. I strongly recommend you keep this selection on.

Previous zoom

If you select previous zoom, you will be asked if you really wanted to do that.

Other zooms

If you select any of the other zooms, you will be asked if you really wanted to do that.

Calc-buf.warn

If you select more than 254 iterations, and you haven't used that many iterations before in that screen mode on that session, you will be asked, if you want to do a whole recalculation. Why this is done, can be read in The basics of Mandela.

HAM8 warn

Mandela can't select the 16.8 million color HAM8 mode of the new AGA Amigas. If you still insist on selecting it, Mandela will default to 4096 color HAM6 to emulate its 256 internal colors. If this warning is set, you will be noticed of the defaulting.

Recalculation

If you select Recalculation, you will be asked if you really wanted to do that.

New

If you select the New, you will be asked if you really wanted to do that.

Quit

If you select the Quit, you will be asked if you really wanted to do that.

1.37 Menu Prefs/Beep after calc. »

Beep after calc. »

Calculating Mandelbrot images can take quite a while. With this option you can make Mandela notify by flashing every screen and beeping when it has finished calculating an image. There are three options you can select:

Always

Beep always when calculation is finished.

If calc >1min

Beep if calculation has taken at least one minute.

Never

Never beep.

1.38 Menu Prefs/Reject modes »

Reject modes »

Reject modes allows you to reject certain screen modes you never want to see in your Screen Mode Requester. The available selections are:

Width < 320: screen modes where display width is ridiculously low

HAM : HAM modes. Recommended if you have AGA graphic chips

All but HAM: all modes that do not use HAM

NTSC : NTSC modes (that are compatible with N-American TV)

PAL : PAL modes (that are compatible with European TV)

DBLNTSC : doublescanned NTSC modes

DBLPAL : doublescanned PAL modes

Multiscan : VGA-compatible multiscan modes

Euro 36 : euro 36 Hz modes

Euro 72 : euro 72 Hz modes

Super 72 : super 72 Hz modes

A2024 : Amiga A2024 monochrome monitor

1.39 Menu Prefs/Color scrolling »

Color scrolling »

Color scrolling lets the user control color scrolling in a Mandelbrot image. This menu is not available if you are running Mandela on an Amiga with an OS older than 3.0, if the screen mode is a HAM mode, or if you have less than 8 colors.

The following color scroll selections are available:

Stop Stops color scrolling and leaves the palette as it is

Forward Starts color scrolling in forward direction

Reverse Starts color scrolling in reverse direction

Faster Makes colors scroll faster

Slower Makes colors scroll slower

Reset Stops color scrolling and restores the palette

1.40 Menu Prefs/Save mode »

Save mode »

Save mode lets the user control how image saving is performed.

The following selections are available:

Screen Everything is saved including headers + info window

Mandelbrot Only the actual Mandelbrot image is saved

1.41 Menu Prefs/Dithering »

Dithering »

Dithering lets the user control how images are displayed.

The following selections are available:

None No dithering

Low Some dithering

High Accurate dithering

Using dithering makes screen updates somewhat slower.

1.42 Menu Prefs/Continuous update ×

Continuous update ×

This switch will select if you want to see all the time Mandela updating its screen when it's calculating an image.

If you clear this selection Mandela will only update your display after it has finished a whole scanline. This may be considerably faster especially with quickly calculated pictures and lots of DMA activity going on.

1.43 Menu Prefs/Load colors w/ images ×

Load colors w/ images ×

This switch lets you select if you want to load a colormap with coordinates.

1.44 Show title bar ×

Show title bar ×

This switch defines if you want the upper screen border to be shown on Mandela's screen.

1.45 Show info window ×

Show info window ×

This switch defines if you want to see an information window that gives you calculation time estimates and other information.

If this window is in your way, you may temporarily drag it to another location. It will, however, come back to its original place when it is updated for the next time.

If you are using a screen larger than your display, you may want to define if the info window is to follow you when you scroll around the virtual screen.

If you are using OS 3.0 or newer, the menus of Mandela are also attached to this window. That way it Mandela's menus work even if you activate the information window.

1.46 Info window follows ×

Info window follows ×

This switch defines if you want to have your information window to follow you if you are using a big autoscrolling screen.

1.47 Menu Zoom

Zoom

Zoom »

In 2X

In 5X

In 10X

Out 2X

Out 5X

Out 10X

Previous

1.48 Menu Zoom/(in and out zooms)

Other zooms

The In 2X, 5X and 10X zooms you in the specified amount. Usually you don't have much use for there.

The Out 2X, 5X and 10X zooms you out the specified amount. These can be used if you feel you are a bit too deep and you can´t go back with the Zoom previous menu selection.

1.49 Menu Zoom/Previous

Previous

This will let you zoom back to the previous position you were in. A 20 zoom history (that is not saved with the preferences or the coordinates) is maintained. There is no way to redo an undoed zoom stage.

1.50 The basics of a Mandelbrot image

The basics of a Mandelbrot image, or $Z=Z^2+C$

Great thanks to Benoit Mandelbrot, who invented this simple formula in 1985. Chaos theory is fun :-)

You know complex numbers? Not? Too bad, because I am not going to explain them here in detail. There are other sources for that. To make it short, complex numbers are 2-dimensional numbers in a way that they have a Real part, and an Imaginary part, whereas the real numbers we use every day only have Real parts.

Now, think of real number values as 2-dimensional coordinates, where the Real part is on the x-axis and the Imaginary part is on the y-axis. This way every point in a 2-dimensional coordinate space has a unique complex value.

Now, to draw a Mandelbrot image we'll do the following iterative operation in pseudo-code for every single point (C is the pixel to be calculated):

INTEGER FUNCTION CALCULATE_POINT(COMPLEX C)

COMPLEX Z = (0.0, 0.0)

INTEGER COUNTER = 0

WHILE (ABS(Z) < 2.0) BEGIN # always true for the first test

$Z = Z^2 + C$

COUNTER = COUNTER + 1

END WHILE

RETURN COUNTER

END FUNCTION

It has been proven (exercise: how?) that if the absolute value of a point exceeds 2.0, it will continue to grow very quickly to explode the limits out of every limit. That also causes the fact that it isn't reasonable to calculate Mandelbrot points for coordinates that has a Real or Imaginary componen that have an absolute value greater than 2.0.

Thus, when the magic limit 2.0 is reached, the iterations stop for that point and the value of COUNTER will determine the color of that pixel. It's that simple!

Although the formula for calculating Mandelbrot images looks deceivingly simple, the image drawn is full of complex patterns. And, it can be zoomed infinitely (see caveats below), and you'll never find precisely the same patterns twice. The exception to this is that the whole Mandelbrot image is mirrored around their x-axis.

Caveats

Some of the coordinates in the Mandelbrot set are of the kind where the absolute value never grows over the magic limit 2.0. Because a program calculating Mandelbrot points would stop forever calculating such a point, an iteration limit must be provided. The pixels that never got a good iteration value are usually colored black. A higher amount of iterations help reduce the black areas somewhat and, naturally, also makes the calculation slower.

Although the Mandelbrot image can theoretically be zoomed infinitely, there are always limits on the precision that a computer can calculate with. That sets a limit to the amount a user can zoom his images. Because integer calculation is usually much faster than floating point calculation, many Mandelbrot programs offer a possibility to calculate images with fixed-point arithmetic simulated with integers. Usually, though, the precision of integers are less than floating point numbers.

The colors used to represent the final image are naturally not set. Many images can be greatly enhanced by giving them exciting colormaps. Also the fact that the color change for every different iteration value, is not enforced nor always desirable. Some programs give the user a definable threshold that lets the user define how many iterations are needed to change the color.

Although, principally, every point in a Mandelbrot image should be calculated to display an image, there are some optimizations that allows programs to get (almost) the exactly correct result with a smaller computational price.

1.51 Optimizations in calculating a Mandelbrot image

Optimizations

The earliest Mandelbrot programs simply calculated every single point in a Mandelbrot image. This was very slow especially if the image contained large black areas.

The first program to use optimizations that I saw, was TurboMandel made in the late 1980's by Philip Marivoet and Nico François.

TurboMandel used a "divide-and-conquer" -method to spare calculations. The idea was to divide the screen in 4 rectangles and calculate all the pixels on the sides of the rectangles. If all of them were of the same color (or, rather, same amount of iterations), the whole rectangle could be painted with that color without calculating the inner pixels of it. If there were pixels of other colors, the rectangle was (recursively) divided into 4 smaller rectangles and the cycle was redone.

This method caused the calculation speed of the black areas to get very much faster.

Later in 1990, Nico François' MandelBlitz used a different approach called contour crawling: it followed the outlines of areas and painted them without calculating any inner pixels at all. This is a bit hard to explain, but since Mandela uses the same algorithm, it is easy for you to check this out.

1.52 Registered users of Mandela

Registered users of Mandela

Thanks go to these nice and honest people who have payed the shareware fee.

950823 Giacomo Mulas $35

950823 Bill Bonner $20

950823 André Schmidt $20

960313 Klaus Eberling $20

960729 Jon Peterson $20

960729 Leo Hansen $20