guide1

COLLABORATORS			
	TITLE : guide1		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 24, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

guide1

1.1 Main Help Index

1.2 Amiga notes

1.3 Requirements

Since this version of fractint uses MUI you will need the mui package that is available on aminet. You will at least need the muimaster.library.

The online amigaguide lookup needs the amigaguide.library (v.33) which is also available on aminet. If it is not present the amigaguide support is disabled.

1.4 Config file

In order to define your own predefined screen resolutions you may copy one of the supplied config files into fractint.cfg (pal users use fractint.cfg.pal or fractint.cfg.AGApal etc.) If fractint finds a fractint.cfg it will be used in its videotable otherwise it will use the builtin table.

The two modes that are not really supported by fractint is EHB and HAM. They actually work (!) well but fractint uses the colors sequentially so smooth transitions from one color to another will produce two ranges (EHB). HAM on the other hand does not have this problem but the colors isnt selected to comply with the HAM standard. The image will have lines going out from a conflicting color pair to the end of the screen. But: if you save the image and it will be stored as a 256 color gif image with the correct colors. Afterwards you may use any giftoham viewer and woila you've got a nice fractal image!

the definition of a screenmode is

```
;key  name of adapter/mode  | AX | BX | CX | DX |mode|  x |  y |clr| comments
;==================================================================================
F1  , PAL lores     ,  11000,        0,   0,   1,   19, 320, 200, 32,
```

```
AX is the screenmode that is sent to OpenScreen. In order to find a new value
you may look in your compilers include files (ex. include:graphics/modeid.h)
or you may use the getmodeid program that comes with xxx.
BX and CX isnt used but the DX value is the Overscan value that is used
by OpenScreen.
```

1.5 Additional contributions

```
Fast 68040 math routines submitted by Roger Uzun (uzun@crash.cts.com)

Several bugs reported by:
Gary Walker
[thanks for the code but I couldnt use it since the diff showed
half the program as different. I have probably used a tabify or someting!.
And comparing line for line in this 56291 line code would take some time!]

Jim Shaffer

The program is so seriously big that trying all options would take
months, so bugreports helps a lot! thanks!
```

1.6 Printing Fractint Documentation

```
You can generate a text file containing full Fractint documentation by
selecting the "Generate FRACTINT.DOC now" hot-link below and pressing
Enter, or by using the DOS command "fractint makedoc=filename" ("filename"
is the name of the file to be written; it defaults to FRACTINT.DOC.)

All information in the documentation file is also available in the online
help, so extracting it is a matter of preference - you can print the
file (e.g.  DOS command "print fractint.doc" or "copy fractint.doc prn")
or read it with a text editor.  It contains over 100 pages of information,
has a table of contents, and is cross-referenced by page number.

  {101 Exit without generating FRACTINT.DOC}

  {100 Generate FRACTINT.DOC now}

Fractint’s great (and unique as far as we know) online help and integrated
documentation file software was written by Ethan Nagel.
```

1.7 new features in version 18.2

```
                Versions 18.1 and 18.2 are bug-fix releases for version 18.0.  ↩
                   Changes from
18.1 to 18.2 include:
```

The <b> command now causes filenames only to be written in PAR files.

Fractint will now search directories in the PATH for files not found in the requested the requested directory or the current directory. If you place .MAP, .FRM, etc. in directories in your PATH, then Fractint will find them.

Fixed bug that caused fractals using PI symmetry to fail at high resolution.

Fractals interrupted with <3> or <r> can now resume.

The palette editor’s <u> (undo) now works.

The <s> command in orbit/Julia window mode is no longer case sensitive.

Added warnings that the POV-Ray output is obsolete (but has been left in). Use POV-Ray’s height field facility instead or create and convert RAW files.

Fixed several IFS bugs.

Changes from 18.0 to 18.1 include:

Overlay tuning - the Mandelbrot/Julia Set fractals are now back up to 17.x speeds

Disk Video modes now work correctly with VESA video adapters (they used to use the same array for different purposes, confusing each other)

1024x768x256 and 2048x2048x256 disk video modes work again

Parameter-file processing no longer crashes Fractint if it attempts to run a formula requiring access to a non-existent FRM file

IFS arrays no longer overrun their array space

type=cellular fixes

"autologmap=2" now correctly picks up the minimum color

The use of disk-video mode with random-access fractal types is now legal (it generates a warning message but lets you proceed if you really want to)

The Lsystems "spinning-wheel" now spins slower (removing needless overhead)

Changes to contributors’ addresses in the Help screens

(The remainder of this "new features" section is from version 18.0)

New fractal types:

19 new fractal types, including:

New fractal types - ’lambda(fn||fn)’, ’julia(fn||fn)’, ’manlam(fn||fn)’, ’mandel(fn||fn)’, ’halley’, ’phoenix’, ’mandphoenix’, ’cellular’, generalized bifurcation, and ’bifmay’ - from Jonathan Osuch.

New Mandelcloud, Quaternion, Dynamic System, Cellular Automata fractal types from Ken Shirriff.

New HyperComplex fractal types from Timothy Wegner

New ICON type from Dan Farmer, including a PAR file of examples.

New Frothy Basin fractal types (and PAR entries) by Wesley Loewer

MIIM (Modified Inverse Iteration Method) implementation of Inverse Julia from Michael Snyder.

New Inverse Julia fractal type from Juan Buhler.

New floating-point versions of Markslambda, Marksmandel, Mandel4, and Julia4 types (chosen automatically if the floating-point option is enabled).

New options/features:

New assembler-based parser logic from Chuck Ebbert - significantly faster than the C-based code it replaces!

New assembler-based Lyapunov logic from Nicholas Wilt and Wes Loewer. Roughly six times faster than the old version!

New Orbits-on-a-window / Julia-in-a-window options:
1) The old Overlay option is now '#' (Shift-3).
2) During generation, 'O' brings up orbits (as before) - after generation, 'O' brings up new orbits Windows mode.
3) Control-O brings up new orbits Windows mode at any time.
4) Spacebar toggles between Inverse Julia mode and the Julia set and back to the Mandelbrot set.

These new "in-a-window" modes are really neat! See
Orbits Window
and Julia Toggle Spacebar Commands for details.

New multi-image GIF support in the <B> command. You can now generate 65535x65535x256 fractal images using Fractint (if you have the disk space, of course). This option builds special PAR entries and a MAKEMIG.BAT file that you later use to invoke Fractint multiple times to generate individual sections of the image and (in a final step) stitch them all together. If your other software can't handle Multiple-image GIFs, a SIMPLGIF program is also supplied that converts MIGS into simgle-image GIFs. Press F1 at the <B> prompts screen for details.

Fractint's decoder now handles Multi-Image Gifs.

New SuperVGA/VESA Autodetect logic from the latest version of VGAKIT. Sure hope we didn't break anything.

New register-compatible 8514/A code from Jonathan Osuch. By default, Fractint now looks first for the presence of an 8514/A register-compatible adapter and then (and only if it doesn't find one) the presence of the 8514/A API (IE, HDILOAD is no longer necessary for register-compatible "8514/a" adapters). Fractint can be forced to use the 8514/A API by using a new command-line option, "afi=yes". Jonathan also added ATI's "8514/a-style" 800x600x256 and 1280x1024x16 modes.

New XGA-detection logic for ISA-based XGA-2 systems.

The palette editor now has a "freestyle" editing option. See

Palette Editing Commands
for details.

Fractint is now more "batch file" friendly. When running Fractint from a batch file, pressing any key will cause Fractint to exit with an errorlevel = 2. Any error that interrupts an image save to disk will cause an exit with errorlevel = 2. Any error that prevents an image from being generated will cause an exit with errorlevel = 1.

New Control-X, Control-Y, and Control-Z options flip a fractal image along the X-axis, Y-axis, and Origin, respectively.

New area calculation mode in TAB screen from Ken Shirriff (for accuracy use inside=0).

The TAB screen now indicates when the Integer Math algorithms are in use.

The palette must now be explicitly changed, it will not reset to the default unexpectedly when doing things like switching video modes.

The Julibrot type has been generalized.
Julibrot fractals can now be generated from PAR files.

Added <b> command support for viewwindows.

Added room for two additional PAR comments in the <B> command

New coloring method for IFS shows which parts of fractal came from which transform.

Added attractor basin phase plotting for Julia sets from Ken Shirriff.

Improved finite attractor code to find more attractors from Ken Shirriff.

New zero function, to be used in PAR files to replace old integer tan, tanh

Debugflag=10000 now reports video chipset in use as well as CPU/FPU type and available memory

Added 6 additional parameters for params= for those fractal types that need them.

New 'matherr()' logic lets Fractint get more aggressive when these errors happen.

New autologmap option (log=+-2) from Robin Bussell that ensures that all palette values are used by searching the screen border for the lowest value and then setting log= to +- that color.

Two new diffusion options - falling and square cavity.

Three new Editpal commands: '!', '@' and '#' commands (that's <shift-1>, <shift-2>, and <shift-3>) to swap R<->G, G<->B, R<->B.

Parameter files now use a slightly shorter maximum line length, making them a bit more readable when stuffed into messages on Compuserve.

Plasma now has 16-bit .POT output for use with Ray tracers. The "old" algorithm has been modified so that the plasma effect is independent of resolution.

Slight modification to the Raytrace code to make it compatible with Rayshade 4.0 patch level 6.

Improved boundary-tracing logic from Wesley Loewer.

Command-line parameters can now be entered on-the-fly using the <g> key thanks to Ken Shirriff.

Dithered gif images can now be loaded onto a b/w display. Thanks to Ken Shirriff.

Pictures can now be output as compressed PostScript. Thanks to Ken Shirriff.

Periodicity is a new inside coloring option. Thanks to Ken Shirriff.

Fixes: symmetry values for the SQR functions, bailout for the floating-pt versions of 'lambdafn' and 'mandelfn' fractals from Jonathan Osuch.

"Flip", "conj" operators are now selectable in the parser

New DXF Raytracing option from Dennis Bragg.

Improved boundary-tracing logic from Wesley Loewer.

New MSC7-style overlay structure is used if MAKEFRAC.BAT specifies MSC7. (with new FRACTINT.DEF and FRACTINT.LNK files for MSC7 users). Several modules have been re-organized to take advantage of this new overlay capability if compiled under MSC7.

Fractint now looks first any embedded help inside FRACTINT.EXE, and then for an external FRACTINT.HLP file before giving up. Previous releases required that the help text be embedded inside FRACTINT.EXE.

Bug fixes:

Corrected formulas displayed for Marksmandel, Cmplxmarksmandel, and associated julia types.

BTM and precision fixes.

Symmetry logic changed for various "outside=" options

Symmetry value for EXP function in lambdafn and lambda(fn||fn) fixed.

```
 Fixed bug where math errors prevented save in batch mode.

 The <3> and <r> commands no longer destroy image -- user can back out
 with ESC and image is still there.

 Fixed display of correct number of Julibrot parameters, and Julibrot
 relaxes and doesn’t constantly force ALTERN.MAP.

 Fixed tesseral type for condition when border is all one color but center
 contains image.

 Fixed integer mandel and julia when used with parameters > +1.99 and < -1.99

 Eliminated recalculation when generating a julia type from a mandelbrot
 type when the ’z’ screen is viewed for the first time.

 Minor logic change to prevent double-clutching into and out of graphics
 mode when pressing, say, the ’x’ key from a menu screen.

 Changed non-US phone number for the Houston Public (Software) Library

 The "Y" screen is now "Extended Options" instead of "Extended Doodads"

 ...and probably a lot more bux-fixes that we’ve since forgotten that
 we’ve implemented.
```

1.8 introduction

FRACTINT plots and manipulates images of "objects" -- actually, sets of mathematical points -- that have fractal dimension.
See Fractals and the PC for some
historical and mathematical background on fractal geometry, a discipline named and popularized by mathematician Benoit Mandelbrot. For now, these sets of points have three important properties:

1) They are generated by relatively simple calculations repeated over and over, feeding the results of each step back into the next -- something computers can do very rapidly.

2) They are, quite literally, infinitely complex: they reveal more and more detail without limit as you plot smaller and smaller areas. Fractint lets you "zoom in" by positioning a small box and hitting <Enter> to redraw the boxed area at full-screen size; its maximum linear "magnification" is over a trillionfold.

3) They can be astonishingly beautiful, especially using PC color displays’ ability to assign colors to selected points, and (with VGA displays or EGA in 640x350x16 mode) to "animate" the images by quickly shifting those color assignments.

For a demonstration of some of Fractint’s features, run the demonstration file included with this release (DEMO.BAT) by typing "demo" at the DOS prompt. You can stop the demonstration at any time by pressing <Esc>.

The name FRACTINT was chosen because the program generates many of its images using INTeger math, rather than the floating point calculations used by most such programs. That means that you don't need a math co-processor chip (aka floating point unit or FPU), although for a few fractal types where floating point math is faster, the program recognizes and automatically uses an 80x87 chip if it's present. It's even faster on systems using Intel's 80386 and 80486 microprocessors, where the integer math can be executed in their native 32-bit mode.

Fractint works with many adapters and graphics modes from CGA to the 1024x768, 256-color XGA mode. Even "larger" images, up to 2048x2048x256, can be plotted to expanded memory, extended memory, or disk: this bypasses the screen and allows you to create images with higher resolution than your current display can handle, and to run in "background" under multi-tasking control programs such as DESQview and Windows 3.

Fractint is an experiment in collaboration. Many volunteers have joined Bert Tyler, the program's first author, in improving successive versions. Through electronic mail messages, first on CompuServe's PICS forum and now on GRAPHDEV, new versions are hacked out and debugged a little at a time. Fractint was born fast, and none of us has seen any other fractal plotter close to the present version for speed, versatility, and all-around wonderfulness. (If you have, tell us so we can steal somebody else's ideas instead of each other's.)
See The Stone Soup Story and A Word About the Authors for information about the authors, and see Contacting the Authors for how to contribute your own ideas and code.

1.9 conditions on use

Fractint is freeware. The copyright is retained by the Stone Soup Group.

Fractint may be freely copied and distributed in unmodified form but may not be sold. (A nominal distribution fee may be charged for media and handling by freeware and shareware distributors.) Fractint may be used personally or in a business - if you can do your job better by using Fractint, or using images from it, that's great! It may not be given away with commercial products without explicit permission from the Stone Soup Group.

There is no warranty of Fractint's suitability for any purpose, nor any acceptance of liability, express or implied.

```
 **********************************************************************
 * Contribution policy: Don't want money. Got money. Want admiration. *
 **********************************************************************
```

Source code for Fractint is also freely available - see
Distribution of Fractint.
See the FRACTSRC.DOC file included with the source for conditions on use.
(In most cases we just want credit.)

1.10 getting started

```
                To start the program, enter FRACTINT at the DOS prompt. The  ↩
                   program
displays an initial "credits" screen. If Fractint doesn't start properly,
please see Common Problems.

Hitting <Enter> gets you from the initial screen to the main menu. You can
select options from the menu by moving the highlight with the cursor arrow
keys and pressing <Enter>, or you can enter commands directly.

As soon as you select a video mode, Fractint begins drawing an image - the
"full" Mandelbrot set if you haven't selected another fractal type.

For a quick start, after starting Fractint try one of the following:
  If you have MCGA, VGA, or better:  <F3>
  If you have EGA:         <F9>
  If you have CGA:         <F5>
  Otherwise, monochrome:       <F6>

After the initial Mandelbrot image has been displayed, try zooming
into it (see
                Zoom Box Commands
                ) and color cycling (see

                Color Cycling Commands
                ).
Once you're comfortable with these basics, start exploring other
functions from the main menu.

Help is available from the menu and at most other points in Fractint by
pressing the <F1> key.

AT ANY TIME, you can hit one of the keys described in
                Display Mode Commands
                a command key to select a function. You do not need to wait for a  ↩
                   calculation
to finish, nor do you have to return to the main menu.

When entering commands, note that for the "typewriter" keys, upper and
lower case are equivalent, e.g. <B> and <b> have the same result.

Many commands and parameters can be passed to FRACTINT as command-line
arguments or read from a configuration file;
see Startup Parameters, Parameter Files for details.
see "Command Line Parameters, Parameter Files, Batch Mode" for details.
```

1.11 display mode commands

```
                Summary of Commands

                Plotting Commands
                Zoom Box Commands

                Image Save/Restore Commands

                Print Command

                Parameter Save/Restore Commands

                Interrupting and Resuming

                Orbits Window

                View Window

                3D Commands

                Video Mode Function Keys

                Hints
```

1.12 Summary of Commands

```
                Hit any of these keys at the menu or while drawing or viewing a fractal.
Commands marked with an '*' are also available at the credits screen.


                Plotting Commands
                 * Delete,F2,F3,.. Select a Video Mode and draw (or redraw) current fractal
 * F1      HELP! (Enter help mode)
   Esc or m    Go to main menu
   \        Redraw previous screen (you can 'back out' recursively)
   Tab       Display information about the current fractal image
 * t       Select a new fractal type and parameters
 * x       Set a number of options and doodads
 * y       Set extended options and doodads
 * z       Set fractal type-specific parameters
   c or + or -     Enter Color-Cycling Mode (see
                Color Cycling Commands
                )
   e       Enter Palette-Editing Mode (see
                Palette Editing Commands
                )
   Spacebar    Mandelbrot/Julia Set toggle.
   Enter     Continue an interrupted calculation (e.g. after a save)
 * f       toggle the floating-point algorithm option ON or OFF
 * i       Set parameters for 3D fractal types
 * Insert    Restart the program (at the credits screen)
   a       Convert the current image into a fractal 'starfield'
   o       toggles 'orbits' option on and off during image generation
 * d       Shell to DOS (type 'exit' at the DOS prompt to return)
   Ctrl-X    Flip the current image along the screen's X-axis
   Ctrl-Y    Flip the current image along the screen's Y-axis
   Ctrl-Z    Flip the current image along the screen's Origin
```

```
               Image Save/Restore Commands
                  s       Save the current screen image to disk
* r       Restore a saved (or .GIF) image ('3' or 'o' for 3-D)


               Orbits Window
                  o       Turns on Orbits Window mode after image generation
  ctrl-o    Turns on Orbits Window mode


               View Window
                * v       Set view window parameters (reduction, aspect ratio)


               Print Command
                  p       Print the screen (command-line options set printer type ↩
                     )


               Parameter Save/Restore Commands
                  b       Save commands describing the current image in a file
      (writes an entry to be used with @ command)
* @       Run a set of commands (in command line format) from a file
  g       Give a startup parameter: Summary of all Parameters


               3D Commands
                * 3       3D transform a saved (or .GIF) image
  # (shift-3)     same as 3, but overlay the current image


               Zoom Box Commands
                  PageUp    When no Zoom Box is active, bring one up
      When active already, shrink it
  PageDown    Expand the Zoom Box
      Expanding past the screen size cancels the Zoom Box
  \24 \25 \27 \26     Pan (Move) the Zoom Box
  Ctrl- \24 \25 \27 \26   Fast-Pan the Zoom Box (may require an enhanced keyboard ↩
     )
  Enter     Redraw the Screen or area inside the Zoom Box
  Ctrl-Enter    'Zoom-out' - expands the image so that your current
      image is positioned inside the current zoom-box location.
  Ctrl-Pad+/Pad-  Rotate the Zoom Box
  Ctrl-PgUp/PgDn  Change Zoom Box vertical size (change its aspect ratio)
  Ctrl-Home/End   Change Zoom Box shape
  Ctrl-Ins/Del    Change Zoom Box color


               Interrupting and Resuming

               Video Mode Function Keys
```

1.13 plotting commands

Function keys & various combinations are used to select a video mode and redraw the screen. For a quick start try one of the following:

If you have MCGA, VGA, or better:	<F3>
If you have EGA:	<F9>
If you have CGA:	<F5>
Otherwise, monochrome:	<F6>

<F1> Display a help screen. The function keys available in help mode are displayed at the bottom of the help screen.

<M> or <Esc> Return from a displayed image to the main menu.

<Esc> From the main menu, <Esc> is used to exit from Fractint.

<Delete> Same as choosing "select video mode" from the main menu. Goes to the "select video mode" screen. See Video Mode Function Keys .

<\> (previously <Home>)
Redraw the previous image. The program tracks 25 sets of previous coordinates and fractal types, but does not remember other options which were different for those past images.

<TAB>
Display the current fractal type, parameters, video mode, screen or (if displayed) zoom-box coordinates, maximum iteration count, and other information useful in keeping track of where you are. The Tab function is non-destructive - if you press it while in the midst of generating an image, you will continue generating it when you return. The Tab function tells you if your image is still being generated or has finished - a handy feature for those overnight, 1024x768 resolution fractal images. If the image is incomplete, it also tells you whether it can be interrupted and resumed. (Any function other than <Tab> and <F1> counts as an "interrupt".)

The Tab screen also includes a pixel-counting function, which will count the number of pixels colored in the inside color. This gives an estimate of the area of the fractal. Note that the inside color must be different from the outside color(s) for this to work; inside=0 is a good choice.

<T>
Select a fractal type. Move the cursor to your choice (or type the first few letters of its name) and hit <Enter>. Next you will be prompted for any parameters used by the selected type - hit <Enter> for the defaults. See Fractal Types for a list of supported types.

<b>
Select a number of basic options. Brings up a full-screen menu of options, any of which you can change at will. These options are:

Passes - see Drawing Method
Floating point toggle - see <F> key description below

```
  Maxiter - see Image Calculation Parameters
  Inside and outside - see Color Parameters
  Savename filename - see File Parameters
  Overwrite option - see File Parameters
  Sound option - see Sound Parameters
  Logmap - see Logarithmic Palettes and Color Ranges
  Biomorph - see Biomorphs
  Decomp - see Decomposition
  Fillcolor - see Drawing Method

<F>
Toggles the use of floating-point algorithms
(see Limitations of Integer Math).
Whether floating point is in
use is shown on the <Tab> status screen.  The floating point option can
also be turned on and off using the "X" options screen.
If you have a non-Intel floating point chip which supports the full 387
instruction set, see the "FPU=" command in Startup Parameters
to get the most out of your chip.

<E>
More options which we couldn't fit under the <b> command:
  "finattract=" - see Finite Attractors
  "potential=" parameters - see Continuous Potential
  "invert=" parameters - see Inversion
  "distest=" parameters - see Distance Estimator Method
  "cyclerange=" - see
                Color Cycling Commands
                <Z>
Modify the parameters specific to the currently selected fractal type.
This command lets you modify the parameters which are requested when you
select a new fractal type with the <T> command, without having to repeat
that selection. You can enter "e" or "p" in column one of the input fields
to get the numbers e and pi (2.71828... and 3.14159...).
From the fractal parameters screen, you can press <F6> to bring up a
sub parameter screen for the coordinates of the image's corners.
; With the IFS fractal type, <Z> brings up the IFS editor (see
; Barnsley IFS Fractals).

<+> or <->
Switch to color-cycling mode and begin cycling the palette
by shifting each color to the next "contour."  See
                Color Cycling Commands
                .
<C>
Switch to color-cycling mode but do not start cycling.
The normally black "overscan" border of the screen changes to white.
See
                Color Cycling Commands
                .

<E>
Enter Palette-Editing Mode.  See
                Palette Editing Commands
                .

<Spacebar>
```

Toggle between Mandelbrot set images and their corresponding Julia-set images. Read the notes in Fractal Types, Julia Sets before trying this option if you want to see anything interesting.

<J>
Toggle between Julia escape time fractal and the Inverse Julia orbit fractal. See Inverse Julias

<Enter>
Enter is used to resume calculation after a pause. It is only necessary to do this when there is a message on the screen waiting to be acknowledged, such as the message shown after you save an image to disk.

<I>
Modify 3D transformation parameters used with 3D fractal types such as "Lorenz3D" and 3D "IFS" definitions, including the selection of funny glasses red/blue 3D.

<A>
Convert the current image into a fractal 'starfield'. See Starfields.

<O> (the letter, not the number)
If pressed while an image is being generated, toggles the display of intermediate results -- the "orbits" Fractint uses as it calculates values for each point. Slows the display a bit, but shows you how clever the program is behind the scenes. (See "A Little Code" in Fractals and the PC.)

<D>
Shell to DOS. Return to Fractint by entering "exit" at a DOS prompt.

<Insert>
Restart at the "credits" screen and reset most variables to their initial state. Variables which are not reset are: savename, lightname, video, startup filename.

1.14 Zoom Box Commands

Zoom Box functions can be invoked while an image is being generated or when it has been completely drawn. Zooming is supported for most fractal types, but not all.

The general approach to using the zoom box is: Frame an area using the keys described below,
then <Enter> to expand what's in the frame to fill the
whole screen (zoom in); or <Ctrl><Enter> to shrink the current image into the framed area (zoom out). With a mouse, double-click the left button to zoom in, double click the right button to zoom out.

<Page Up>, <Page Down>
Use <Page Up> to initially bring up the zoom box. It starts at full screen size. Subsequent use of these keys makes the zoom box smaller or larger. Using <Page Down> to enlarge the zoom box when it is already at maximum size removes the zoom box from the display. Moving the mouse away from you or toward you while holding the left button down performs the same

functions as these keys.

Using the cursor "arrow" keys or moving
the mouse without holding any buttons down, moves the zoom box.

Holding <Ctrl> while pressing cursor "arrow" keys moves the box 5 times faster. (This only works with enhanced keyboards.)

Panning: If you move a fullsize zoombox and don't change anything else before performing the zoom, Fractint just moves what's already on the screen and then fills in the new edges, to reduce drawing time. This feature applies to most fractal types but not all. A side effect is that while an image is incomplete, a full size zoom box moves in steps larger than one pixel. Fractint keeps the box on multiple pixel boundaries, to make panning possible. As a multi-pass (e.g. solid guessing) image approaches completion, the zoom box can move in smaller increments.

In addition to resizing the zoom box and moving it around, you can do some rather warped things with it. If you're a new Fractint user, we recommend skipping the rest of the zoom box functions for now and coming back to them when you're comfortable with the basic zoom box functions.

<Ctrl><Keypad->, <Ctrl><Keypad+>
Holding <Ctrl> and pressing the numeric keypad's + or - keys rotates the zoom box. Moving the mouse left or right while holding the right button down performs the same function.

<Ctrl><Page Up>, <Ctrl><Page Down>
These commands change the zoom box's "aspect ratio", stretching or shrinking it vertically. Moving the mouse away from you or toward you while holding both buttons (or the middle button on a 3-button mouse) down performs the same function. There are no commands to directly stretch or shrink the zoom box horizontally - the same effect can be achieved by combining vertical stretching and resizing.

<Ctrl><Home>, <Ctrl><End>
These commands "skew" the zoom box, moving the top and bottom edges in opposite directions. Moving the mouse left or right while holding both buttons (or the middle button on a 3-button mouse) down performs the same function. There are no commands to directly skew the left and right edges - the same effect can be achieved by using these functions combined with rotation.

<Ctrl><Insert>, <Ctrl><Delete>
These commands change the zoom box color. This is useful when you're having trouble seeing the zoom box against the colors around it. Moving the mouse away from you or toward you while holding the right button down performs the same function.

You may find it difficult to figure out what combination of size, position rotation, stretch, and skew to use to get a particular result. (We do.) A good way to get a feel for all these functions is to play with the Gingerbreadman fractal type. Gingerbreadman's shape makes it easy to see what you're doing to him. A warning though: Gingerbreadman will run forever, he's never quite done! So, pre-empt with your next zoom when he's baked enough.

If you accidentally change your zoom box shape or rotate and forget which way is up, just use <PageDown> to make it bigger until it disappears, then <PageUp> to get a fresh one. With a mouse, after removing the old zoom box from the display release and re-press the left button for a fresh one.

If your screen does not have a 4:3 "aspect ratio" (i.e. if the visible display area on it is not 1.333 times as wide as it is high), rotating and zooming will have some odd effects - angles will change, including the zoom box's shape itself, circles (if you are so lucky as to see any with a non-standard aspect ratio) become non-circular, and so on. The vast majority of PC screens *do* have a 4:3 aspect ratio.

Zooming is not implemented for the plasma and diffusion fractal types, nor for overlayed and 3D images. A few fractal types support zooming but do not support rotation and skewing - nothing happens when you try it.

1.15 Image Save/Restore Commands

<S> saves the current image to disk. All parameters required to recreate the image are saved with it. Progress is marked by colored lines moving down the screen's edges.

The default filename for the first image saved after starting Fractint is FRACT001.GIF; subsequent saves in the same session are automatically incremented 002, 003... Use the "savename=" parameter or <b> options screen to change the name. By default, files left over from previous sessions are not overwritten - the first unused FRACTnnn name is used. Use the "overwrite=yes" parameter or <b> options screen) to overwrite existing files.

A save operation can be interrupted by pressing any key. If you interrupt, you'll be asked whether to keep or discard the partial file.

<R> restores an image previously saved with <S>, or an ordinary GIF file. After pressing <R> you are shown the file names in the current directory which match the current file mask. To select a file to restore, move the cursor to it (or type the first few letters of its name) and press <Enter>.

Directories are shown in the file list with a "" at the end of the name. When you select a directory, the contents of that directory are shown. Or, you can type the name of a different directory (and optionally a different drive) and press <Enter> for a new display. You can also type a mask such as "*.XYZ" and press <Enter> to display files whose name ends with the matching suffix (XYZ).

You can use <F6> to switch directories to the default fractint directory or to your own directory which is specified through the DOS environment variable "FRACTDIR".

Once you have selected a file to restore, a summary description of the file is shown, with a video mode selection list. Usually you can just press <Enter> to go past this screen and load the image. Other choices available at this point are:

```
  Cursor keys: select a different video mode
  <Tab>: display more information about the fractal
  <F1>: for help about the "err" column in displayed video modes
```

If you restore a file into a video mode which does not have the same pixel dimensions as the file, Fractint will make some adjustments: The view window parameters (see <V> command) will automatically be set to an appropriate size, and if the image is larger than the screen dimensions, it will be reduced by using only every Nth pixel during the restore.

1.16 print command

<P>

Print the current fractal image on your (Laserjet, Paintjet, Epson-compatible, PostScript, or HP-GL) printer.

See Setting Defaults (SSTOOLS.INI File) and Printer Parameters for how to let Fractint know about your printer setup.

Disk-Video can be used to generate images for printing at higher resolutions than your screen supports.

1.17 Parameter Save/Restore Commands

Parameter files can be used to save/restore all options and settings required to recreate particular images. The parameters required to describe an image require very little disk space, especially compared with saving the image itself.

<@>

The <@> command loads a set of parameters describing an image. (Actually, it can also be used to set non-image parameters such as SOUND, but at this point we’re interested in images. Other uses of parameter files are discussed in Parameter Files and the <@> Command.)

When you hit <@>, Fractint displays the names of the entries in the currently selected parameter file. The default parameter file, FRACTINT.PAR, is included with the Fractint release and contains parameters for some sample images.

After pressing <@>, highlight an entry and press <Enter> to load it, or press <F6> to change to another parameter file.

Note that parameter file entries specify all calculation related parameters, but do not specify things like the video mode - the image will be plotted in your currently selected mode.

<B>

The <B> command saves the parameters required to describe the currently displayed image, which can subsequently be used with the <@> command to recreate it.

After you press <B>, Fractint prompts for:

Parameter file: The name of the file to store the parameters in. You should use some name like "myimages" instead of fractint.par, so that your images are kept separate from the ones released with new versions of Fractint. You can use the PARMFILE= command in SSTOOLS.INI to set the default parameter file name to "myimages" or whatever. (See Setting Defaults (SSTOOLS.INI File) and "parmfile=" in File Parameters.)

Name: The name you want to assign to the entry, to be displayed when the <@> command is used.

Main comment: A comment to be shown beside the entry in the <@> command display.

Second, Third, and Fourth comment: Additional comments to store in the file with the entry. These comments go in the file only, and are not displayed by the <@> command.

Record colors?: Whether color information should be included in the entry. Usually the default value displayed by Fractint is what you want. Allowed values are:
"no" - Don’t record colors. This is the default if the image is using your video adapter’s default colors.
"@mapfilename" - When these parameters are used, load colors from the named color map file. This is the default if you are currently using colors from a color map file.
"yes" - Record the colors in detail. This is the default when you’ve changed the display colors by using the palette editor or by color cycling. The only reason that this isn’t what Fractint always does for the <B> command is that color information can be bulky - up to nearly 1K of disk space. That may not sound like much, but can add up when you consider the thousands of wonderful images you may find you just *have* to record... Smooth-shaded ranges of colors are compressed, so if that’s used a lot in an image the color information won’t be as bulky.

# of colors: This only matters if "Record colors?" is set to "yes". It specifies the number of colors to record. Recording less colors will take less space. Usually the default value displayed by Fractint is what you want. You might want to increase it in some cases, e.g. if you are using a 256 color mode with maxiter 150, and have used the palette editor to set all 256 possible colors for use with color cycling, then you’ll want to set the "# of colors" to 256.

At the bottom of the input screen are inputs for Fractint’s "pieces" divide-and-conquer feature. You can create multiple PAR entries that break an image up into pieces so that you can generate the image pieces one by one. There are two reasons for doing this. The first is in case the fractal is very slow, and you want to generate parts of the image at the same time on several computers. The second is that you might want to make an image greater than 2048 x 2048. The parameters for this feature are:
X Multiples - How many divisions of final image in the x direction

```
     Y Multiples - How many divisions of final image in the y direction
     Video mode  - Fractint video mode for each piece (e.g. "F3")
```

The last item defaults to the current video mode. If either X Multiples or Y Multiples are greater than 1, then multiple numbered PAR entries for the pieces are added to the PAR file, and a MAKEMIG.BAT file is created that builds all of the component pieces and then stitches them together into a "multi-image" GIF. The current limitations of the "divide and conquer" algorithm are 36 or fewer X and Y multiples (so you are limited to "only" 36x36=1296 component images), and a final resolution limit in both the X and Y directions of 65,535 (a limitation of "only" four billion pixels or so).

The final image generated by MAKEMIG is a "multi-image" GIF file called FRACTMIG.GIF. In case you have other software that can't handle multi-image GIF files, MAKEMIG includes a final (but commented out) call to SIMPLGIF, a companion program that reads a GIF file that may contain little tricks like multiple images and creates a simple GIF from it. Fair warning: SIMPLGIF needs room to build a composite image while it works, and it does that using a temporary disk file equal to the size of the final image - and a 64Kx64K GIF image requires a 4GB temporary disk file!

<G>

The <G> command lets you give a startup parameter interactively.

1.18 <b>asic Options Screen

```
   Passes - see Drawing Method
   Fillcolor - see Drawing Method
   Floating Point Algorithm - see notes below
   Maximum Iterations - see Image Calculation Parameters
   Inside and Outside colors - see Color Parameters
   Savename and File Overwrite - see File Parameters
   Sound option - see Sound Parameters
   Log Palette - see Logarithmic Palettes and Color Ranges
   Biomorph Color - see Biomorphs
   Decomp Option - see Decomposition
```

You can toggle the use of floating-point algorithms on this screen (see Limitations of Integer Math). Whether floating point is in use is shown on the <Tab> status screen. If you have a non-Intel floating point chip which supports the full 387 instruction set, see the "FPU=" command in Startup Parameters to get the most out of your chip.

1.19 <E>xtended Options Screen

```
                  Finite attractor - seeFinite Attractors

   Potential parameters - seeContinuous Potential
```

Distance Estimator parameters - seeDistance Estimator Method

Inversion parameters - seeInversion

Color cycling range - see
Color Cycling Commands

1.20 Image Coordinates Screen

You can directly enter corner coordinates on this screen instead of using the zoom box to move around. You can also use <F4> to reset the coordinates to the defaults for the current fractal type.

There are two formats for the display: corners or center-mag. You can toggle between the two by using <F7>.

In corners mode, corner coordinate values are entered directly. Usually only the top-left and bottom-right corners need be specified - the bottom left corner can be entered as zeros to default to an ordinary unrotated rectangular area. For rotated or skewed images, the bottom left corner must also be specified.

Center-mag mode can only be used with unrotated unstretched images. In this mode the image area is described by entering the coordinates for the center of the rectangle, and its magnification factor.

1.21 Interrupting and Resuming

Fractint command keys can be loosely grouped as:

- o Keys which suspend calculation of the current image (if one is being calculated) and automatically resume after the function. <Tab> (display status information) and <F1> (display help), are the only keys in this group.

- o Keys which automatically trigger calculation of a new image. Examples: selecting a video mode (e.g. <F3>); selecting a fractal type using <T>; using the <b> screen to change an option such as maximum iterations.

- o Keys which do something, then wait for you to indicate what to do next. Examples: <M> to go to main menu; <C> to enter color cycling mode; <PageUp> to bring up a zoom box. After using a command in this group, calculation automatically resumes when you return from the function (e.g. <Esc> from color cycling, <PageDn> to clear zoom box). There are a few fractal types which cannot resume calculation, they are noted below. Note that after saving an image with <S>, you must press <Enter> to clear the "saved" message from the screen and resume.

An image which is <S>aved before it completes can later be <R>estored and continued. The calculation is automatically resumed when you restore such an image.

When a slow fractal type resumes after an interruption in the third category above, there may be a lag while nothing visible happens. This is because most cases of resume restart at the beginning of a screen line. If unsure, you can check whether calculation has resumed with the <Tab> key.

The following fractal types cannot (currently) be resumed: plasma, 3d transformations, julibrot, and 3d orbital types like lorenz3d. To check whether resuming an image is possible, use the <Tab> key while it is calculating. It is resumable unless there is a note under the fractal type saying it is not.

The Batch Mode section discusses how to resume in batch mode.

To <R>estore and resume a "formula", "lsystem", or "ifs" type fractal your "formulafile", "lfile", or "ifsfile" must contain the required name.

1.22 Orbits Window

The <O> key turns on the Orbit mode. In this mode a cursor appears over the fractal. A window appears showing the orbit used in the calculation of the color at the point where the cursor is. Move the cursor around the fractal using the arrow keys or the mouse and watch the orbits change. Try entering the Orbits mode with View Windows (<V>) turned on. The following keys take effect in Orbits mode.

<c> Circle toggle - makes little circles with radii inversely proportional to the iteration. Press <c> again to toggle back to point-by-point display of orbits.

<l> Line toggle - connects orbits with lines (can use with <c>)

<n> Numbers toggle - shows coordinates of the cursor on the screen. Press <n> again to turn off numbers.

<p> Enter pixel coordinates directly

<h> Hide fractal toggle. Works only if View Windows is turned on and set for a small window (such as the default size.) Hides the fractal, allowing the orbit to take up the whole screen. Press <h> again to uncover the fractal.

<s> Saves the fractal, cursor, orbits, and numbers as they appear on the screen.

<<> or <,> Zoom orbits image smaller

<>> or <.> Zoom orbits image larger

<z> Restore default zoom.

1.23 View Window

The <V> command is used to set the view window parameters described below. These parameters can be used to:

- o Define a small window on the screen which is to contain the generated images. Using a small window speeds up calculation time (there are fewer pixels to generate). You can use a small window to explore quickly, then turn the view window off to recalculate the image at

full screen size.
- Generate an image with a different "aspect ratio"; e.g. in a square window or in a tall skinny rectangle.
- View saved GIF images which have pixel dimensions different from any mode supported by your hardware. This use of view windows occurs automatically when you restore such an image.

"Preview display"
Set this to "yes" to turn on view window, "no" for full screen display. While this is "no", the only view parameter which has any affect is "final media aspect ratio". When a view window is being used, all other Fractint functions continue to operate normally - you can zoom, color-cycle, and all the rest.

"Reduction factor"
When an explicit size is not given, this determines the view window size, as a factor of the screen size. E.g. a reduction factor of 2 makes the window 1/2 as big as the screen in both dimensions.

"Final media aspect ratio"
This is the height of the final image you want, divided by the width. The default is 0.75 because standard PC monitors have a height:width ratio of 3:4. E.g. set this to 2.0 for an image twice as high as it is wide. The effect of this parameter is visible only when "preview display" is enabled.

"Crop starting coordinates"
This parameter affects what happens when you change the aspect ratio. If set to "no", then when you change aspect ratio, the prior image will be squeezed or stretched to fit into the new shape. If set to "yes", the prior image is "cropped" to avoid squeezing or stretching.

"Explicit size"
Setting these to non-zero values over-rides the "reduction factor" with explicit sizes in pixels. If only the "x pixels" size is specified, the "y pixels" size is calculated automatically based on x and the aspect ratio.

More about final aspect ratio: If you want to produce a high quality hard-copy image which is say 8" high by 5" down, based on a vertical "slice" of an existing image, you could use a procedure like the following. You'll need some method of converting a GIF image to your final media (slide or whatever) - Fractint can only do the whole job with a PostScript printer, it does not preserve aspect ratio with other printers.
- restore the existing image
- set view parameters: preview to yes, reduction to anything (say 2), aspect ratio to 1.6, and crop to yes
- zoom, rotate, whatever, till you get the desired final image
- set preview display back to no
- trigger final calculation in some high res disk video mode, using the appropriate video mode function key
- print directly to a PostScript printer, or save the result as a GIF file and use external utilities to convert to hard copy.

1.24 3d commands

See 3D Images for details of these commands.

<3>
Restore a saved image as a 3D "landscape", translating its color information into "height". You will be prompted for all KINDS of options.

<#>
Restore in 3D and overlay the result on the current screen.

1.25 Video Mode Function Keys

Fractint supports *so* many video modes that we've given up trying to reserve a keyboard combination for each of them.

Any supported video mode can be selected by going to the "Select Video Mode" screen (from main menu or by using <Delete>), then using the cursor up and down arrow keys and/or <PageUp> and <PageDown> keys to highlight the desired mode, then pressing <Enter>.

Up to 39 modes can be assigned to the keys F2-F10, SF1-SF10 <Shift>+<Fn>), CF1-CF10 (<Ctrl>+<Fn>), and AF1-AF10 (<Alt>+<Fn>). The modes assigned to function keys can be invoked directly by pressing the assigned key, without going to the video mode selection screen.

30 key combinations can be reassigned: <F1> to <F10> combined with any of <Shift>, <Ctrl>, or <Alt>.
The video modes assigned to <F2> through <F10> can not be changed - these are assigned to the most common video modes, which might be used in demonstration files or batches.

To reassign a function key to a mode you often use, go to the "select video mode" screen, highlight the video mode, press the keypad (gray) <+> key, then press the desired function key or key combination. The new key assignment will be remembered for future runs.

To unassign a key (so that it doesn't invoke any video mode), highlight the mode currently selected by the key and press the keypad (gray) <-> key.

A note about the "select video modes" screen:
the video modes which are displayed with a 'B' suffix in the number of colors are modes which have no custom programming - they use the BIOS and are S-L-O-W ones.

See Video Adapter Notes for comments about particular adapters.

See Disk-Video for a description of these non-display modes.

See Customized Video Modes, FRACTINT.CFG for information about adding your own video modes.

1.26 hints

Remember, you do NOT have to wait for the program to finish a full screen display before entering a command. If you see an interesting spot you want to zoom in on while the screen is half-done, don't wait -- do it! If you think after seeing the first few lines that another video mode would look better, go ahead -- Fractint will shift modes and start the redraw at once. When it finishes a display, it beeps and waits for your next command.

In general, the most interesting areas are the "border" areas where the colors are changing rapidly. Zoom in on them for the best results. The first Mandelbrot-set (default) fractal image has a large, solid-colored interior that is the slowest to display; there's nothing to be seen by zooming there.

Plotting time is directly proportional to the number of pixels in a screen, and hence increases with the resolution of the video mode. You may want to start in a low-resolution mode for quick progress while zooming in, and switch to a higher-resolution mode when things get interesting. Or use the solid guessing mode and pre-empt with a zoom before it finishes. Plotting time also varies with the maximum iteration setting, the fractal type, and your choice of drawing mode. Solid-guessing (the default) is fastest, but it can be wrong: perfectionists will want to use dual-pass mode (its first-pass preview is handy if you might zoom pre-emptively) or single-pass mode.

When you start systematically exploring, you can save time (and hey, every little bit helps -- these "objects" are INFINITE, remember!) by <S>aving your last screen in a session to a file, and then going straight to it the next time by using the command FRACTINT FRACTxxx (the .GIF extension is assumed), or by starting Fractint normally and then using the <R> command to reload the saved file. Or you could hit <B> to create a parameter file entry with the "recipe" for a given image, and next time use the <@> command to re-plot it.

1.27 fractint on unix

Fractint has been ported to Unix to run under X Windows. This version is called "Xfractint". Xfractint may be obtained by anonymous ftp to sprite.Berkeley.EDU, in the file xfractnnn.shar.Z.

Xfractint is still under development and is not as reliable as the IBM PC version.

Contact Ken Shirriff (shirriff@cs.Berkeley.EDU) for information on Xfractint.

Xfractint is a straight port of the IBM PC version. Thus, it uses the IBM user interface. If you do not have function keys, or Xfractint does not accept them from your keyboard, use the following key mappings:

IBM	Unix
F1 to F10	Shift-1 to Shift-0

INSERT	I
DELETE	D
PAGE_UP	U
PAGE_DOWN	N
LEFT_ARROW	H
RIGHT_ARROW	L
UP_ARROW	K
DOWN_ARROW	J
HOME	O
END	E
CTL_PLUS	}
CTL_MINUS	{

Xfractint takes the following options:

-onroot
Puts the image on the root window.

-fast
Uses a faster drawing technique.

-disk
Uses disk video.

-geometry
Changes the geometry of the image window.

-display displayname
Specifies the X11 display to use.

-private
Allocates the entire colormap (i.e. more colors).

-share
Shares the current colormap.

-fixcolors n
Uses only n colors.

-slowdisplay
Prevents xfractint from hanging on the title page with slow displays.

-simple
Uses simpler keyboard handling, which makes debugging easier.

Common problems:

If you get the message "Couldn’t find fractint.hlp", you can
a) Do "setenv FRACTDIR /foo", replacing /foo with the directory containing fractint.hlp.
b) Run xfractint from the directory containing fractint.hlp, or
c) Copy fractint.hlp to /usr/local/bin/X11/fractint

If you get the message "Invalid help signature", the problem is due to byteorder. You are probably using a Sun help file on a Dec machine or vice versa.

If xfractint doesn't accept input, try typing into both the graphics window and the text window. On some systems, only one of these works.

If you are using Openwindows and can't get xfractint to accept input, add to your .Xdefaults file:

```
OpenWindows.FocusLenience:      True
```

If you cannot view the GIFs that xfractint creates, the problem is that xfractint creates GIF89a format and your viewer probably only handles GIF87a format. Run "xfractint gif87a=y" to produce GIF87a format.

Because many shifted characters are used to simulate IBM keys, you can't enter capitalized filenames.

1.28 Color Cycling Commands

See
Color Cycling Command Summary
for a summary of commands.

Color-cycling mode is entered with the 'c', '+', or '-' keys from an image, or with the 'c' key from Palette-Editing mode.

The color-cycling commands are available ONLY for VGA adapters and EGA adapters in 640x350x16 mode. You can also enter color-cycling while using a disk-video mode, to load or save a palette - other functions are not supported in disk-video.

Note that the colors available on an EGA adapter (16 colors at a time out of a palette of 64) are limited compared to those of VGA, super-VGA, and MCGA (16 or 256 colors at a time out of a palette of 262,144). So color-cycling in general looks a LOT better in the latter modes. Also, because of the EGA palette restrictions, some commands are not available with EGA adapters.

Color cycling applies to the color numbers selected by the "cyclerange=" command line parameter (also changeable via the <E> options screen and via the palette editor). By default, color numbers 1 to 255 inclusive are cycled. On some images you might want to set "inside=0" (<b> options or command line parameter) to exclude the "lake" from color cycling.

When you are in color-cycling mode, you will either see the screen colors cycling, or will see a white "overscan" border when paused, as a reminder that you are still in this mode. The keyboard commands available once you've entered color-cycling. are described below.

<F1>
Bring up a HELP screen with commands specific to color cycling mode.

<Esc>
Leave color-cycling mode.

<+> or <->
Begin cycling the palette by shifting each color to the next "contour."
<+> cycles the colors in one direction, <-> in the other.

'<' or '>'
Force a color-cycling pause, disable random colorizing, and single-step through a one color-cycle. For "fine-tuning" your image colors.

Cursor up/down
Increase/decrease the cycling speed. High speeds may cause a harmless flicker at the top of the screen.

<F2> through <F10>
Switches from simple rotation to color selection using randomly generated color bands of short (F2) to long (F10) duration.

<1> through <9>
Causes the screen to be updated every 'n' color cycles (the default is 1). Handy for slower computers.

<Enter>
Randomly selects a function key (F2 through F10) and then updates ALL the screen colors prior to displaying them for instant, random colors. Hit this over and over again (we do).

<Spacebar>
Pause cycling with white overscan area. Cycling restarts with any command key (including another spacebar).

<Shift><F1>-<F10>
Pause cycling and reset the palette to a preset two color "straight" assignment, such as a spread from black to white. (Not for EGA)

<Ctrl><F1>-<F10>
Pause & set a 2-color cyclical assignment, e.g. red->yellow->red (not EGA).

<Alt><F1>-<F10>
Pause & set a 3-color cyclical assignment, e.g. green->white->blue (not EGA).

<R>, <G>, <B>
Pause and increase the red, green, or blue component of all colors by a small amount (not for EGA). Note the case distinction of this vs:

<r>, <g>, <b>
Pause and decrease the red, green, or blue component of all colors by a small amount (not for EGA).

<D> or <A>
Pause and load an external color map from the files DEFAULT.MAP or ALTERN.MAP, supplied with the program.

<L>
Pause and load an external color map (.MAP file). Several .MAP files are supplied with Fractint. See Palette Maps.

<S>
Pause, prompt for a filename, and save the current palette to the named file (.MAP assumed). See Palette Maps.

1.29 color cycling command summary

```
                  See
                Color Cycling Commands
                  F1       HELP! (Enter help mode and display this screen)
  Esc      Exit from color-cycling mode
  + or -     (re)-set the direction of the color-cycling
  27 26        (re)-set the direction of the color-cycling (just like +/-)
  24 25        SpeedUp/SlowDown the color cycling process
  Right/Left Arrow (re)-set the direction of the color-cycling (just like +/-)
  Up/Down Arrow    SpeedUp/SlowDown the color cycling process
  F2 thru F10    Select Short--Medium--Long (randomly-generated) color bands
  1  thru 9    Cycle through 'nn' colors between screen updates (default=1)
  Enter      Randomly (re)-select all new colors  [TRY THIS ONE!]
  Spacebar     Pause until another key is hit
  < or >     Pause and single-step through one color-cycle
* SF1 thru AF10    Pause and reset the Palette to one of 30 fixed sequences
  d or a     pause and load the palette from DEFAULT.MAP or ALTERN.MAP
  l      load palette from a map file
  s      save palette to a map file
* r or g or b or   force a pause and Lower (lower case) or Raise (upper case)
* R or G or B    the Red, Green, or Blue component of the fractal image
```

1.30 Palette Editing Commands

See
Palette Editing Command Summary
for a summary of commands.

Palette-editing mode provides a number of tools for modifying the colors in an image. It can be used only with MCGA or higher adapters, and only with 16 or 256 color video modes.
Many thanks to Ethan Nagel for creating the palette editor.

Use the <E> key to enter palette-editing mode from a displayed image or from the main menu.

When this mode is entered, an empty palette frame is displayed. You can use the cursor keys to position the frame outline, and <Pageup> and <Pagedn> to change its size. (The upper and lower limits on the size depend on the current video mode.) When the frame is positioned where you want it, hit Enter to display the current palette in the frame.

Note that the palette frame shows R(ed) G(reen) and B(lue) values for two color registers at the top. The active color register has a solid frame, the inactive register's frame is dotted. Within the active register, the active color component is framed.

Using the commands described below, you can assign particular colors to the registers and manipulate them. Note that at any given time there are two colors "X"d - these are pre-empted by the editor to display the palette frame. They can be edited but the results won't be visible. You can change which two colors are borrowed ("X"d out) by using the <v> command.

Once the palette frame is displayed and filled in, the following commands are available:

<F1>
Bring up a HELP screen with commands specific to palette-editing mode.

<Esc>
Leave palette-editing mode

<H>
Hide the palette frame to see full image; the cross-hair remains visible and all functions remain enabled; hit <H> again to restore the palette display.

Cursor keys
Move the cross-hair cursor around. In ’auto’ mode (the default) the color under the center of the cross-hair is automatically assigned to the active color register. Control-Cursor keys move the cross-hair faster. A mouse can also be used to move around.

<R> <G> <B>
Select the Red, Green, or Blue component of the active color register for subsequent commands

<Insert> <Delete>
Select previous or next color component in active register

<+> <->
Increase or decrease the active color component value by 1 Numeric keypad (gray) + and - keys do the same.

<Pageup> <Pagedn>
Increase or decrease the active color component value by 5; Moving the mouse up/down with left button held is the same

<0> <1> <2> <3> <4> <5>
Set the active color component’s value to 0 10 20 ... 60

<Space>
Select the other color register as the active one. In the default ’auto’ mode this results in the now-inactive register being set to remember the color under the cursor, and the now-active register changing from whatever it had previously remembered to now follow the color.

<,> <.>
Rotate the palette one step. By default colors 1 through 255 inclusive are rotated. This range can be over-ridden with the "cyclerange" parameter, the <E> options screen, or the <O> command described below.

"<" ">"
Rotate the palette continuously (until next keystroke)

<O>
Set the color cycling range to the range of colors currently defined by the color registers.

<C>
Enter Color-Cycling Mode. When you invoke color-cycling from here, it will subsequently return to palette-editing when you <Esc> from it. See

Color Cycling Commands

.

<=>
Create a smoothly shaded range of colors between the colors selected by the two color registers.

<M>
Specify a gamma value for the shading created by <=>.

<D>
Duplicate the inactive color register's values to the active color register.

<T>
Stripe-shade - create a smoothly shaded range of colors between the two color registers, setting only every Nth register. After hitting <T>, hit a numeric key from 2 to 9 to specify N. For example, if you press <T> <3>, smooth shading is done between the two color registers, affecting only every 3rd color between them. The other colors between them remain unchanged.

<W>
Convert current palette to gray-scale. (If the <X> or <Y> exclude ranges described later are in force, only the active range of colors is converted to gray-scale.)

<Shift-F2> ... <Shift-F9>
Store the current palette in a temporary save area associated with the function key. The temporary save palettes are useful for quickly comparing different palettes or the effect of some changes - see next command. The temporary palettes are only remembered until you exit from palette-editing mode.

<F2> ... <F9>
Restore the palette from a temporary save area. If you haven't previously saved a palette for the function key, you'll get a simple grey scale.

<L>
Pause and load an external color map (.MAP file). See Palette Maps.

<S>
Pause, prompt for a filename, and save the current palette to the named file (.MAP assumed). See Palette Maps.

<I>
Invert frame colors. With some colors the palette is easier to see when the frame colors are interchanged.

<>
Move or resize the palette frame. The frame outline is drawn - it can then be repositioned and sized with the cursor keys, <Pageup> and <Pagedn>, just as was done when first entering palette-editing mode. Hit

```
Enter when done moving/sizing.

<V>
Use the colors currently selected by the two color registers for the
palette editor’s frame.  When palette editing mode is entered, the last
two colors are "X"d out for use by the palette editor; this command can be
used to replace the default with two other color numbers.

<A>
Toggle ’auto’ mode on or off.  When on (the default), the active color
register follows the cursor; when off, <Enter> must be pressed to set the
active register to the color under the cursor.

<Enter>
Only useful when ’auto’ is off, as described above; double clicking the
left mouse button is the same as Enter.

<X>
Toggle ’exclude’ mode on or off - when toggled on, only those image pixels
which match the active color are displayed.

<Y>
Toggle ’exclude’ range on or off - similar to <X>, but all pixels matching
colors in the range of the two color registers are displayed.

<N>
Make a negative color palette - will convert only current color if in ’x’
mode or range between editors in ’y’ mode or entire palette if in "normal"
mode.

<!>
<@>
<#>
Swap R<->G, G<->B, and R<->B columns. These keys are shifted 1, 2, and 3,
which you may find easier to remember.

<U>
Undoes the last palette editor command.  Will undo all the way to the
beginning of the current session.

<E>
Redoes the undone palette editor commands.

<F>
Toggles "Freestyle mode" on and off (Freestyle mode changes a range of
palette values smoothly from a center value outward).
With your cursor inside the palette box, press the <F> key to enter
Freestyle mode.  A default range of colors will be selected for you
centered at the cursor (the ends of the color range are noted by putting
dashed lines around the corresponding palette values). While in Freestyle
mode:

 Moving the mouse changes the location of the range of colors that are
 affected.

 Control-Insert/Delete or the shifted-right-mouse-button changes the
 size of the affected palette range.
```

```
The normal color editing keys (R,G,B,1-6, etc) set the central color
of the affected palette range.

Pressing ENTER or double-clicking the left mouse button makes the
palette changes permanent (if you don’t perform this step, any
palette changes disappear when you press the <F> key again to exit
freestyle mode).
```

1.31 palette editing command summary

```
               ; This topic is online only.

 See
               Palette Editing Commands
                for full documentation.

 F1       HELP! (Enter help mode and display this screen)
 Esc      Exit from palette editing mode
 h      Hide/unhide the palette frame
 Cursor keys    Move the cross-hair cursor around. Control-Cursor keys
      move faster. A mouse can also be used to move around.
 r or g or b    Select the the Red, Green, or Blue component of the
      active color register for subsequent commands
 Insert or Delete Select previous or next color component in active register
 + or -     Increase or decrease the active color component by 1
 Pageup or Pagedn Increase or decrease the active color component by 5;
      Moving the mouse up/down with left button held is the same
 0 1 2 3 4 5 6    Set active color component to 0 10 20 ... 60
 Space      Select the other color register as the active one
 , or .     Rotate the palette one step
 < or >     Rotate the palette continuously (until next keystroke)
 c      Enter Color-Cycling Mode (see
               Color Cycling Commands
               )
 =      Create a smoothly shaded range of colors
 m      Set the gamma value for ’=’.
 d      Duplicate the inactive color register in active color
 t      Stripe-shade; after hitting ’t’, hit a number from 2 to 9
      which is used as stripe width
 Shift-F2,F3,..F9 Store the current palette in a temporary save area
      associated with the function key
 F2,F3,...,F9     Restore the palette from a temporary save area
 w      Convert palette (or current exclude range) to gray-scale
 \       Move or resize the palette frame
 i      Invert frame colors, useful with dark colors
 a      Toggle ’auto’ mode on or off - when on, the active color
      register follows the cursor; when off, Enter must be hit
      to set the register to the color under the cursor
 Enter      Only useful when ’auto’ is off, as described above; double
      clicking the left mouse button is the same as Enter
 x      Toggle ’exclude’ mode on or off
 y      Toggle ’exclude’ range on or off
 o      Set the ’cyclerange’ (range affected by color cycling
      commands) to the range of the two registers
```

```
  n      Make a negative color palette
  u      Undoes the last command
  e      Redoes the last undone command
  !      Swap red and green columns
  @      Swap green and blue columns
  #      Swap red and blue columns
  f               Enter Palette-Editing Mode.  See
                Palette Editing Commands
                                   for details.
```

1.32 Terje Pedersen

If you want to contact me you can reach me on
internet email terjepe@stud.cs.uit.no
or by mail:

Terje Pedersen
Oterveien 55
9017 Tromsø
Norway

I am currently studying and pretty much broke, and soon just very broke! So if you find yourself using this program a lot: Please feel free to send a small contribution to encourage me to continue improving this wonderful program! The stone soup group don't seem to want any money, but I'm not one of them! ($5-$10 sounds like a resonable amount, any amount above this is also amazingly welcome..)

```
 *********************************************************************
 * Contribution policy: Don't want money. Got money. Want admiration. *
 *********************************************************************
```

1.33 Mui interface

This application uses

MUI - MagicUserInterface

(c) Copyright 1993 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 20.- or US$ 15.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

1.34 Future plans

There are no future plans!

1.35 Speed of amiga version of fractint

Here is a couple of news articles that came up on sci.fractals.
It may be a result of an error somewhere but until anyone finds the
error I find it highly amusing!

(I love to harass PC'owners (My mission in life in fact..:-))

```
---
Article: 3516 of sci.fractals
Path: news.uit.no!trane.uninett.no!sunic!columba.udac.uu.se!meryl.csd.uu.se!patsve
From: patsve@meryl.csd.uu.se (Patrik Svensson)
Newsgroups: sci.fractals
Subject: Re: Speed of fractint
Date: 28 Jun 1994 15:29:25 GMT
Organization: Uppsala University
Lines: 48
Message-ID: <2upfkl$m3f@columba.udac.uu.se>
References: <1994Jun23.110408.9688@news.uit.no> <1994Jun27.150132.4639@ericsson.se> <1994Jun28.120618.28492@news.uit.no>
NNTP-Posting-Host: meryl.csd.uu.se

terjepe@stud.cs.uit.no (Terje Pedersen) writes:

>In article <1994Jun27.150132.4639@ericsson.se>, etxmtha@ericsson.se (Mattias Hartikainen) writes:
>|> In article 9688@news.uit.no, terjepe@stud.cs.uit.no (Terje Pedersen) writes:
>|> >In the dgeneral.par file to fractint I found an entry:
>|> >
>|> >minimandel         { ; took 10 hours on 486 (using copro)
>|> >                     ; not worth it, either
>|> >
>|> >I tried it on the amiga version on my A4000/040 and it took
>|> >1 and a half minute!
>|> >also the:
>|> >minimandel3        { ; took 10 hours on a 486, and it needs copro.
>|> >                     ; not worth the time otherwise.
>|> >
>|> >It took about 3 minutes!
>|> >
>|> >Can this really be true?
>|> >Isn’t the 486 supposed to be roughly as fast as the 68040?
>|> >
>|> >(the resolution I used was 360x280x256)
>|> >--
>|> >TP
>|>
>|> Are you sure you used the same settings on both machines?
>|> Of greatest importance would be some such as the maximum number
>|> of iterations, the colouring technique (1-/2-pass, guessing,
>|> boundary tracing) etc. It seems strange otherwise; even if the
>|> 68040 is faster...

>the par file :
>  reset type=mandel
>  corners=-1.786105797595/-1.786105544097/0.000091122677/0.000091312687
>  float=y maxiter=5000 logmap=yes

>So it sets the maxiter and the other options is reset to default if I’m not
>totally mistaken.

>I think the 68040 is supposed to be slightly faster than the 486 on
>math operations but 10 times as fast sound slightly more..

>|> /Mattias

>--
>TP

On my 486/66 i took (minimandel 360x280x256) 7:49.94, hardly 10 hours but
not 1 and a half minute either.

----

Article: 3487 of sci.fractals
Path: news.uit.no!trane.uninett.no!sunic!EU.net!howland.reston.ans.net!cs.utexas.edu!news.tamu.edu!waldorf.tamu.edu!not-for-mail
From: loewer@waldorf.tamu.edu (Wes)
Newsgroups: sci.fractals
Subject: Re: Speed of fractint
Date: 24 Jun 1994 08:07:10 -0500
Organization: Texas A&M University, College Station
Lines: 21
Message-ID: <2uelpu$8qb@waldorf.tamu.edu>
References: <1994Jun23.110408.9688@news.uit.no>
NNTP-Posting-Host: waldorf.tamu.edu

In article <1994Jun23.110408.9688@news.uit.no>,
Terje Pedersen <terjepe@stud.cs.uit.no> wrote:
>I tried it on the amiga version on my A4000/040 and it took
>1 and a half minute!

>It took about 3 minutes!
```

> Can this really be true?
> Isn't the 486 supposed to be roughly as fast as the 68040?
>
> (the resolution I used was 360x280x256)

At this resolution, they took 15:31.09 and 25:20.39 respectively on my 486DX-33. Certainly appears to be a considerable difference. Taking in to consideration that the ibm version uses hand coded assembler floating point routines, this is especially big.
I don't know if the amiga version uses the C code or amiga assembler. I do know that the using the 80x86 assembler is about twice as fast as the C code on an ibm. I wonder how this compares to a Pentium?

-wes loewer

1.36 Why this SLOW version?

The main reason why I am not releasing the full speed versions of Fractint this time is that currently I have only received 1 request/contribution (for a total of $5..) and it was slightly below expectation. I reasoned that since you already have the full version, why waste extra money? But this time around you don't have the full version (in terms of speed) and maybe it will be what makes you actually send me a contribution. Maybe there is no money to be gained from fractal programs.
If that is the case this may well be my last Fractint version for the Amiga. (Next release of the PC version of Fractint will then only be runnable with a PC emulator on the Amiga .., until someone else find it worth the effort to port it..) Not that I will promise to port newer versions of Fractint as they become available, but at least there will be a chance I will do so. (if it turns out to be worthwhile)

It's all up to you!

TP

1.37 Still problems?

If you think Fractint behaves in strange ways on your computer, it may well be true! To check some initial stuff that fractint does (like checking for os3.0) you can add a command line option 'debug=2' and a couple of lines of info will be printed. Also if you are running with a gfx card you could try 'force' and the program will use 8 bitplanes instead of 6. (It may have discovered you have an ECS machine).

The documentation is HUGE, and somewhere in there, there may be an answer to your question..

TP