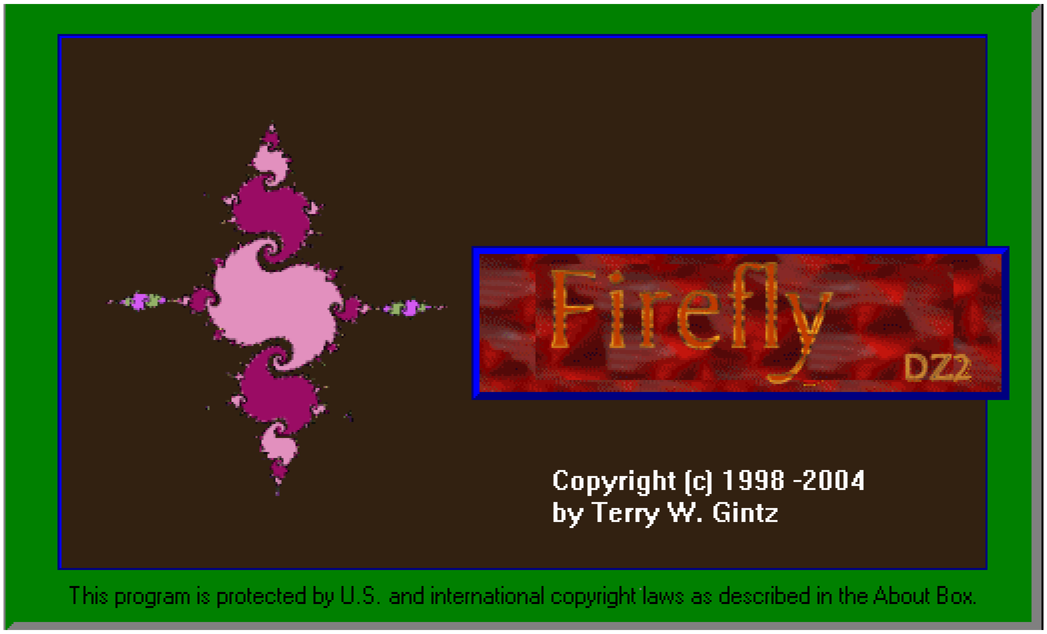


Firefly Application Help

Table of Contents

1 Main Index

Firefly Help Index

Getting Started

Firefly Remote

Commands

File menu
Edit menu
Image menu
Type menu
Map menu
Break menu
Render menu
Pixel menu
Color menu
Convolve menu
View menu
Window menu
Audio/Video menu
Demo menu
Help menu

1.1 Title Bar

Title Bar

The title bar is located along the top of a window. It contains the name of the application and drawing.

To move the window, drag the title bar. Note: You can also move dialog boxes by dragging their title bars.

A title bar may contain the following elements:

- Application Control-menu button
- Drawing Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Name of the drawing
- Restore button

1.2 Scroll Bars

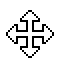
Scroll bars

Displayed at the right and bottom edges of the drawing window. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the drawing. You can use the mouse to scroll to other parts of the drawing.

1.3 Size

Size command (System menu)

Use this command to display a four-headed arrow so you can size the active window with the arrow keys.



After the pointer changes to the four-headed arrow:
1. Press one of the DIRECTION keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a DIRECTION key to move the border.
3. Press ENTER when the window is the size you want.

Note: This command is unavailable if you maximize the window.

Shortcut
Mouse: Drag the size bars at the corners or edges of the window.

1.4 Move

Move command (Control menu)

Use this command to display a four-headed arrow so you can move the active window or dialog box with the arrow keys.

Note: This command is unavailable if you maximize the window.

Shortcut
Keys: CTRL+F7

1.5 Minimize Command

Minimize command (application Control menu)

Use this command to reduce the Firefly window to an icon.

Shortcut

Mouse: Click the minimize icon on the title bar.
Keys: ALT+F9

1.6 Maximize Command

Maximize command (System menu)

Use this command to enlarge the active window to fill the available space.

Shortcut

Mouse: Click the maximize icon on the title bar; or double-click the title bar.
Keys: CTRL+F10 enlarges a drawing window.

1.7 Next Window

Next Window command (drawing Control menu)

Use this command to switch to the next open drawing window. Firefly determines which window is next according to the order in which you opened the windows.

Shortcut
Keys: CTRL+F6

1.8 Previous Window

Previous Window command (drawing Control menu)

Use this command to switch to the previous open drawing window. Firefly determines which window is previous according to the order in which you opened the windows.

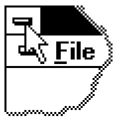
Shortcut
Keys: SHIFT+CTRL+F6

1.9 Close

Close command (Control menus)

Use this command to close the active window or dialog box.

Double-clicking a Control-menu box is the same as choosing the Close command.



Shortcuts

Keys: CTRL+F4 closes a drawing window
ALT+F4 closes the application

1.10 Restore

Restore command (Control menu)

Use this command to return the active window to its size and position before you chose the Maximize or Minimize command.

1.11 Switch to

Switch to command (application Control menu)

Use this command to display a list of all open applications. Use this "Task List" to switch to or close an application on the list.

Shortcut

Keys: CTRL+ESC

Dialog Box Options
When you choose the Switch To command, you will be presented with a dialog box with the following options:

Task List
Select the application you want to switch to or close.

Switch To
Makes the selected application active.

End Task
Closes the selected application.

Cancel
Closes the Task List box.

Cascade
Arranges open applications so they overlap and you can see each title bar. This option does not affect applications reduced to icons.

Tile
Arranges open applications into windows that do not overlap. This option does not affect applications reduced to icons.
Arrange Icons
Arranges the icons of all minimized applications across the bottom of the screen.

2 Firefly Remote

Firefly Remote

The remote provides access to many of the most-used commands in Firefly. Information about each button can be obtained by using the '?' located near the close box in the top right-hand corner.

2.1 New

New button

Use this button to create a new drawing window in Firefly.

2.2 Undo

Undo button

Use this command to undo the last action. An image can be continued after an undo, if continue was enabled before the last action, but not after an image is resized. Color-cycling is disabled after using Undo.

2.3 Size

Size button

This allows you to set the drawing area for a picture, independent of the Windows screen size. It also shows which size is currently in use. The aspect for the drawing is based on the ratio of X (horizontal width) to Y (vertical height.) The custom setting allows for any size/aspect that system memory will permit. The minimum size for an image is 40X30. Note: if the image is less than 100 width, the aspect must be 4/3 for solid-guessing to work properly.

2.4 Color

Color button

Use the palette editor to modify the palette(s) in use.

2.5 Batch

Batch button

Here you set parameters for random scanning, batching and saving random-generated images to disk.

2.6 Fvr

FVR button

The window opened contains all the major variables that Firefly now scales between key frames of an avi stream.

2.7 Draw button

Draw button

Use this button to draw or redraw the image for the current fractal variables. Clicking inside the draw window with the left-mouse button stops all plotting. Use the Cont button to restart plotting from the current column.

2.8 Abort button

Abort button

Use this command to stop drawing. Clicking inside a window's drawing area or close box (or the program close box) will also stop the drawing. Note: once a plot has started Firefly continues to draw the image for that window regardless of which drawing window has the input focus, until done or aborted. You can open and close other drawing windows without affecting the current drawing, but only one drawing is active at any time.

2.9 View

View button

Displays the entire plot, expanding or shrinking the image to fit in a maximized window without title bar, scroll bars or menu bar. At all other times, part of the picture is hidden by the inclusion of the title bar, toolbar, scroll bars and menu bar. To exit full-screen mode, press any key or click the left-mouse button.

2.10 Scan

Scan button

This generates a Julia set or quaternion Julia set from a formula's Mandelbrot 'P' space. Random points in a formula's current Mandelbrot space are scanned for an interesting Julia set. Rendering options are maintained in the current fractal. Equivalent to the 'F' hot key.

2.11 Form

Form button

Use this button to change formulas or type. The window opened varies with the type of formula in use (plug-in or built-in.)

2.12 Help

Help button

Use this button to open the help index for Firefly.

2.13 Channel JL

Random Julia (Channel JL button)

A random Julia fractal is generated. The Mandelbrot space for one of the plug-in formulas is scanned for an interesting Julia set. The plug-in rendering and filter used are also randomized (when random coloring and rendering is enabled in the Batch window.) Note: In most case the Julia search is a short one, but sometimes the "seek" mode can seem to get stuck when the criteria for an interesting Julia set fails to match the formula used. In the latter case, either click the left mouse button and restart the search process.

Tip: some things remain to be done after the Julia set is drawn. The algorithm doesn't check the type of split palette that is used, so a mismatch in the "Divide by" color option may need correcting, e.g. Divide-by-four may be selected for a divide-by-eight palette. Feel free to experiment with all the parameters, reframe the image, change palettes etc. This routine provides a fast intro to many options in Firefly that the user may be unfamiliar with: no knowledge of fractal science/math required!

2.14 Channel RN

Random Render (Channel RN button)

The rendering is randomized using available options for normal or generalized coloring modes. This include the Invert and Torus options.

2.15 Channel CN

Random Coloring (Channel CN button)

The coloring is randomized using available options for normal or generalized coloring modes. This include the Blend and RGB options.

2.16 Channel JB

Random Julia (Channel JB button)

A random Julia fractal is generated. Many of the rendering options of Firefly are selected on a random basis, and the Mandelbrot space for one of the built-in formulas is scanned for an interesting Julia set. The palette used is also randomized (when random coloring and rendering is enabled in the Batch window.). Note: In most case the Julia search is a short one, but sometimes the "seek" mode can seem to get stuck when the criteria for an interesting

Julia set fails to match the formula used. In the latter case, either click the left mouse button and restart the search process.

Tip: some things remain to be done after the Julia set is drawn. The algorithm doesn't check the type of split palette that is used, so a mismatch in the "Divide by" color option may need correcting, e.g. Divide-by-four may be selected for a divide-by-eight palette. Feel free to experiment with all the parameters, reframe the image, change palettes etc. This routine provides a fast intro to many options in Firefly that the user may be unfamiliar with: no knowledge of fractal science/math required!

2.17 Channel RP

Random Render2 (Channel RP button)

The rendering is randomized using available options for the palette-based mode. This include most of the built-in rendering options.

2.18 Channel CP

Random Coloring2 (Channel CP button)

The palette is randomized and coloring mode set to palette-based.

2.19 Load

Load button

Use this button to open an existing data/image file in a new window. You can open multiple image files at once. Use the Window menu to switch among the multiple open images.

2.20 Save

Save button

Use this button to save and name the active drawing. Firefly displays the Save As dialog box so you can name your drawing.

To save a drawing with its existing name and directory, use the File/Save command.

2.21 Bmp

BMP button

Use this button to select the BMP format when loading and saving fractals. This is the default Windows bitmap format, readable by most Windows programs that use image files. This is also the fastest method of loading and saving fractals, but requires more disk space, since no compression is used. Windows keeps track of the last six BMP files saved or loaded (displayed in the Files menu.)

2.22 Png

PNG radio button

Use this button to select the PNG format when loading and saving fractals. This format uses medium compression without loss of image quality.

2.23 Jpg

JPG radio button

Use this button to select the JPEG format when loading and saving fractals. This format uses moderate compression but with some loss of image quality. This is preferable for posting to the net, since most browsers can display jpeg files.

2.24 |||||

||||| button

Through a series of windows, this allows you to name and open an avi animation stream and choose a compression method. After choosing the frame rate (1-60 fps) and using the file requester to name the file, you are given a choice of compression methods. You can also choose no compression for optimum view quality. (All compression methods degrade the original images, some more than others.) The first key frame in the stream is then drawn and written to the file.

Note: after the stream is opened, the size of the fractal that can be drawn is fixed at the size of the frame. No changes can be made to the size until the stream is closed.

2.25 >

> button

With this option, frames are written to a stream based on the difference between the current key frame and the previous key frame. The first key frame is written when you open a stream. The next key frame is created each time you use this option. In between you can zoom or change Fvr variables as much as necessary. The stream is only written to when this option is used. The last key frame is automatically saved after the 'tween' series is written. The number of frames may range from 1-1500 frames between keys. With a frame number of 1 only the key frames are written. This allows animation to be created that incorporate all scalable variables in FF.

Use the Cancel button to exit this dialog without initializing a new series of frames.

Check the Log Scaling box if you want the frames to be written with logarithmic space between frames, else linear space is used. Useful when zooming, where frames would otherwise be packed together at the end of the frame series.

2.26 []

[ ] button

Closes any open avi stream file. You need to do this before viewing the file or creating a new avi file. The stream is also closed when you exit Firefly.

2.27 V

V button

Opens an avi file for viewing. You can preview any multimedia file by clicking on its file name. A multimedia box will appear to the right of the file list. Click on okay to open the main view window.

There are buttons to Play a file forwards or Backwards, or forward automatically with Auto rewind/repeat. Click on Slow to slow down a video. Each click on Slow halves the viewing speed. A click on Stop freezes viewing and restores the view speed to normal playback.

Use the Open button to view a different avi file. Use the Save button to save the file in a different compression format. You must use a different name to save the file than the name that was used to open it. Click on the left-mouse button or any key to abort a save operation.

Note: the view avi requester can be used to preview any multimedia file, including midi files.

3 File menu

File menu commands

The File menu offers the following commands:

New	Creates a new drawing.
Open [BMP+DZ/ZP]	Opens an existing drawing.
Close	Closes an opened drawing.
Save [BMP+DZ]	Saves an opened drawing using the same file name.
Save As [BMP+DZ]	Saves an opened drawing to a specified file name.
Save Parameters [DZ]	Save parameters for an opened drawing to a specified file name.
Load Parameters [DZ/ZP]	Load parameters from an existing drawing.
Load Palettes [PL]	Load palettes file.
Save Palettes [PL]	Save palettes to file.
Load Palette [MAP]	Load a Fractint map file.
Open [JPEG]	Load jpeg.
Save As [JPEG]	Save in jpeg format.
Open [PNG]	Load png.
Save As [PNG]	Save in png format.
Exit	Exit Firefly.

3.1 File New command

New command (File menu)

Use this command to create a new drawing window in Firefly. The image and data for the opening picture are used to create the new window.

You can open an existing data/image file with the Open command.

Shortcuts
Keys: CTRL+N

3.2 File Open command

Open command (File menu)

Use this command to open an existing data/image file in a new window. You can open multiple image files at once. Use the Window menu to switch among the multiple open images. See Window 1, 2, ... command.

You can create new images with the New command.

Shortcuts
Toolbar: 
Keys: CTRL+O

3.2.1 File Open dialog box

File Open dialog box

The following options allow you to specify which file to open:

File Name
Type or select the filename you want to open. This box lists files with the extension you select in the List Files of Type box.

List Files of Type
Select the type of file you want to open:

Drives
Select the drive in which Firefly stores the file that you want to open.

Directories
Select the directory in which Firefly stores the file that you want to open.

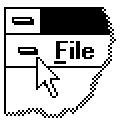
Network...
Choose this button to connect to a network location, assigning it a new drive letter.

3.3 File Close command

Close command (File menu)

Use this command to close the window containing the active image. If you close a window without saving, you lose all changes made since the last time you saved it.

You can also close a drawing by using the Close icon on the drawing window, as shown below:



3.4 File Save command

Save command (File menu)

Use this command to save the active drawing to its current name and directory. When you save a drawing for the first time, Firefly displays the Save As dialog box so you can name your drawing. If you want to change the name and directory of an existing drawing before you save it, choose the Save As command.

Shortcuts

Toolbar: 
Keys: CTRL+S

3.5 File Save As command

Save As command (File menu)

Use this command to save and name the active drawing. Firefly displays the Save As dialog box so you can name your drawing.

To save a drawing with its existing name and directory, use the Save command.

3.5.1 File Save as dialog box

File Save As dialog box

The following options allow you to specify the name and location of the file you're about to save:

File Name

Type a new filename to save a drawing with a different name. Firefly adds the extension .dz.

Drives
Select the drive in which you want to store the drawing.
Directories
Select the directory in which you want to store the drawing.
Network...
Choose this button to connect to a network location, assigning it a new drive letter.

3.6 File Load Parameters command

Load Parameters command (File menu)

Use this command to load a data file [.dz]. The data file contains all variables to recreate an image created previously with Firefly. Note: for Firefly , the default data extension is "dz", but data files with the zp extension can also be loaded using the "All files" option in the files type.

3.7 File Save Parameters command

Save Parameters command (File menu)

Use this command to save all data elements for the current image in a data file [.dz].

3.8 File Load Palettes command

Load Palettes command (File menu)

Use this command to load a palette file [.pl]. The palette file contains 21 palettes created previously with Firefly (or another version of the program.)

3.9 File Save Palettes command

Save Palettes command (File menu)

Use this command to save all palettes for the current session in a palette file [.pl].

3.10 File Load Palette command

Load Palette [MAP] command (File menu)

Use this command to load a Fractint-type map file. The palette file replaces the palette in F12.

3.11 File Open[JPG] command

Open [JPEG] command (File menu)

Use this command to load parameters and a bitmap file that were saved in jpeg format. There is an option in the file-type box to load only the bitmap too. This replaces the Open command for those who need a smaller sized bitmap file. Note: the last files list doesn't keep track of images loaded in JPEG format. Note: for Firefly , the default data extension is "dz", but data files with the zp extension can also be loaded using the "All files" option in the files type.

3.12 File Save Bitmap As [JPG] command

Save As [JPEG] command (File menu)

Use this command to save the parameters and active bitmap in jpeg format. There is an option in the file-type box to save only the bitmap too. This replaces the Save and Save As command for those who need a smaller sized bitmap file. Note: the last files list doesn't keep track of images saved in JPEG format.

3.13 File Open [PNG] command

Open [PNG] command (File menu)

Use this command to load parameters and a bitmap file that was saved in png format. There is an option in the file-type box to load only the bitmap too. This replaces the Open command for those who need a smaller sized bitmap file. Note: the last files list doesn't keep track of images loaded in PNG format. Note: for Firefly , the default data extension is "dz", but data files with the zp extension can also be loaded using the "All files" option in the files type.

3.14 File Save Bitmap As [PNG] command

Save As [PNG] command (File menu)

Use this command to save the parameters and active bitmap in png format. There is an option in the file-type box to save only the bitmap too. This replaces the Save and Save As command for those who need a smaller sized bitmap file. Note: the last files list doesn't keep track of images saved in PNG format.

3.15 File 1, 2, 3, 4, 5, 6 command

1, 2, 3, 4, 5, 6 command (File menu)

Use the numbers and filenames listed at the bottom of the File menu to open the last six drawings you closed. Choose the number that corresponds with the drawing you want to open.

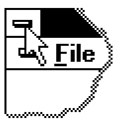
3.16 File Exit command

Exit command (File menu)

Use this command to end your Firefly session. You can also use the Close command on the application Control menu. Note: if you choose to exit while plotting, the program does not terminate, but stops the plotting so the program can be safely exited.

Shortcuts

Mouse: Double-click the application's Control menu button.



Keys: ALT+F4

4 Edit menu

Edit menu commands

The Edit menu offers the following commands:

Undo	Undo last edit, action or zoom.
Copy	Copy the active view and put it on the Clipboard.
Copy	Define area of view and copy to clipboard.
Paste	Insert Clipboard contents.
Copy Data	Copy fractal data to buffer.
Paste Data	Copy data from copy buffer.
Built-in Formulas	Edit built-in formula/type data.
Fractal Parameters	Edit fractal parameters.
Plug-in Configuration	Edit plug-in formula/rendering list.
Fractint Variables/Options	Edit Fractint-compatible variables/options.
Size	Sets the image size.
RGB Thresholds	Set threshold values for color interpolation.
Text	Edit and add text to drawing.

4.1 Edit Undo command

Undo command (Edit menu)

Use this command to undo the last action. An image can be continued after an undo, if continue was enabled before the last action. Color-cycling is disabled after using Undo, though.

Shortcut
Keys: CTRL+Z

4.2 Edit Copy command

Copy command (Edit menu)

Use this command to copy the active view to the clipboard. The entire view is copied to the clipboard.

Shortcut
Keys: CTRL+C

4.3 Edit Clip command

Clip command (Edit menu)

Use this command to copy a part of the active view to the clipboard. A zoom box is used to select the part to be copied. Click outside the view frame or press escape to exit this option.

Shortcut
Keys: CTRL+L

4.4 Edit Paste command

Paste command (Edit menu)

Use this command to paste from the clipboard. The clipboard must contain a bitmap. If the bitmap is larger than the view, it is clipped. The zoom cursor is used to set the left/top corner in the view where the bitmap will be pasted. Click outside the view frame or press escape to exit this option. Note: using this command in conjunction with the Merge Color options, you can blend two or more bitmaps together in various ways, using Boolean rules. This doesn't necessarily produce the same affect as using the Merge options to draw over another fractal.

Shortcut

Keys: CTRL+V

4.5 Edit Copy Data command

Copy Data command (Edit menu)

Use this command to copy the fractal data for the active view to the file "c:\zcopy.zp". The current palette for the view is also copied.

Shortcut
Keys: CTRL+F

4.6 Formula Window

Built-in Formula/Type Window
Fun #1 and Fun #2 are combo controls for entering/selecting up to two formulas in the form of Z=AZ+BZ+c. Z is the complex variable or function, 'c' is the complex constant, and A and B are optional real constants. There is additionally a Type control, an Arg control and an Arg Limit control that determine how the above formulas are processed.

The Type control accepts a value of 0 to 11. For a value of 0, the first formula is always used and the second formula is ignored. Dll action lists are only used with Type 0 and Type 11.

For a value of 1, the second formula is processed and the first formula is ignored. Type 1 is of use only if you are switching between two functions and don't want to reenter them each time you plot the other one.

For a value of 2, the first formula is processed if the Arg value (another formula which must contain only real terms) is greater than or equal to the arg limit set with the Limit gadget.

For values of 3, the first formula is processed and its output becomes the input of the second formula, which is then processed. For fractals that are non-convergent types, this produces a composite function (a function of a function.) For Newton types and other convergent fractals, this produces a union of both functions. Convergence may occur with either of the functions (if both are convergent.)

Type 4 takes the average of Fun#1 and Fun#2.

Type 5 alternates between the two functions while iterating.

Type 6 takes the quotient of both functions.

Types 7 and 8 are exploratory feedback types using the results of two formulas run consecutively. Type 7 is an 'or' function, where formula 1 and formula 2 are iterated from the same starting point, and the results of one formula is used for coloring purposes. The criteria used for selecting which formula to use differs depending on if the first formula is convergent or not. If the first formula is convergent, then if its convergent time is less than the escape or convergent time of the second formula then use the first formula's results, else keep the last formula's results. If the first formula is non-convergent, then if its escape time is greater than that of the second formula then use the first formula's results, else keep the last formula's results.

Type 8 pipes the results of the first formula into the second formula and uses the results of the second formula.

Type 9 uses the Formula box to enter up to 1000 characters per formula.
Text can be pasted from the clipboard to the formula box by using the keystrokes shift-insert.
Text may be moved from box to box by using shift-delete to move it first to the clipboard.

Type 10 produces Escher-like tilings for Julia sets, as described in The Science of Fractal Images. This uses both fun#1 and fun#2 in combination. Fun#1 sets the stage for the target set fun#2. To reproduce the images in The Science of Fractal Images, use z=z*z for fun#1 and p0 for fun#2. Fun#1 may be any formula other than z=z*z, but may produce unsymmetrical tilings. Fun#2 can be any Julia formula, convergent or non-convergent. Hypercomplex extensions are also supported, though an overflow condition may cause the tiling to revert to its non-hypercomplex version. This type is mainly applicable to Julia-type sets, but may be used with limited effect with the MandelbrotP-type set. Use the Rise box in the Parameters window to set the degree of tiling desired. The higher the Rise number (1-235), the greater the density of the tiling.

Type 11 is like type 9, except that the formula is used only with a plug-in action list. This allows you to bypass built-in rendering methods, which might interfere with a plug-in rendering action, and still use your own custom formula.

When you select a function from the first 100 formulas (P0-F9) in fun #1 list box, a thumbnail picture appears at the lower left of the dialog window. This shows some of the aspects of the formula chosen. (To reduce file space for the sample data in Firefly, bitmaps for the thumbnail are displayed as 16-color bitmaps rather than 256-color or true-color bitmaps.) Click on the Use Example button to transfer the data that created the picture to the current image data. (The 4-bit palette in the thumbnail is not transferred.) You can then see how the picture was done, draw it full-screen, zoom on it and/or play with its parameters to create new versions of the picture. Note: the examples were originally created with ZPlot (the 256-color program), so may appear differently when drawn with Firefly. The color-scaling factor (cutoff) may have to be decreased tenfold or more, and the palette rotated, to see a similar picture.

To accommodate the Arg function, there are special variables, which the program treats as strictly real values. 'X' is the real value of Z. 'Y' is the coefficient of the imaginary part of 'Z'. 'R' is the absolute value of Z. 'J' is real part of the complex constant. 'K' is the coefficient of the imaginary part of the complex constant. These and other real variables described below can be used in the Arg input, to create IFS maps of complex functions, as described by

Michael Barnsley in his book, Fractals Everywhere.

Note: Using the Arg gadget for type 2 functions is unnecessary when using one of the built-in IFS functions. The Arg gadget is used in built-in functions to specify the display method for Newton/convergence functions. For more info on how to set Newton/Halley/Renormalization flags with the Arg gadget, see the notes at the end of built-in formulas section.

The f1-f4 combo boxes are used to designate the 'fn' user-definable part of a generalized function. This can be one of the 41 function types listed (sin(w),cos(w) etc.) A few of the options are formulas themselves (L3(w) the Legendre function, the gamma function and G(w) the Gaussian function), so quite complex (though mathematically unintelligible) formulas are possible.

The S control is used to enter the variable 's' used in many of the built-in functions.
The Si control is used to enter the variable 'si' the imaginary component for s in some of the built-in functions.
The Converge control is used to enter the convergence limit for Newton/Renormalization/Convergence plots. This is a measure for how close z needs to approach an attractor to be mapped to that attractor color. A small value .0001-.000001 is normally used. Smaller values of the convergence variable produce a more accurate plot, while increasing the computation time somewhat. On some formulas it may be necessary to reduce this variable to its smallest value (.00000001) to eliminate some artifacts (spots) caused by non-convergence (at a higher limit value.)

About formula syntax: This applies if you elect to enter your own formula into one of the function boxes and use the parser to generate the plot. The use of parenthesis is necessary around complex exponents or variables. E.g.: '(z-i)^(1.5e)'. For a complete list of variables, operators and functions recognized by the parser, see Parser Information.

Up to 500 user-named-complex variables and constants may be included in a formula. A variable must begin with a letter and may contain numbers and letters only. A variable may be up to 9 characters long. A constant may be up to 20 digits long, including the decimal point. Firefly uses syntax similar to Fractint's formula style with an initialization section, followed by the main formula, and an optional bailout routine. Comments may be entered on the same lime with a preceding ';'. Some variables such as 'pixel' and p1 are named after Fractint's predefined variables. These are provided to allow Firefly users to more easily convert Fractint formula types to Firefly use. However, Firefly doesn't prompt you to enter values into p1 (the cr and ci boxes) or p2 (the s and si boxes) or p3 (the limit and converge boxes). Since p1 is used in the iteration process as 'c', p1 cannot be used as a variable independent of c. A ':' terminates the initialization section. Multiple phrases may be entered in the main formula or initialization sections on the same line by using the terminator ',' between phrases. Use ctrl-enter to terminate a line in the formula box. An optional bailout routine may be entered as a phrase at the end of the formula. If the bailout phrase equals a value other than TRUE during iteration, the iteration loop is exited. There are other flags such as Convergence and Biomorph, if set, that can force exit also.
The arg control is limited to 25 characters.

The Title text box is used with the hot key 'T' to annotate a picture with text. Use the Edit/Text command to change font, text color or format text into multiple lines. Text in this box is not saved in a picture's data file, but once entered the same text can be used over and over for different pictures. Useful for adding copyright/author info to batches of pictures. Since the same title text may be used many times, it is shared among views and saved in the file "prefs.txt" in Firefly's startup directory.

Click on the Okay button to use the formulas currently displayed in the window, or Cancel to exit the window without making any changes.

The Reset button returns all boxes and slider values to their original values when the window was opened.

The three buttons named Rand fun#1, Rand fun#2 and Rand f1-f4 are used to pick formulas/functions at random. Clicking on Rand fun#1, a formula is chosen (from the 100 built-in formulas) for fun #1. Clicking on Rand fun#2, a formula is chosen for fun #2. Clicking on Rand f1-f4, functions are chosen for f1-f4.

4.6.1 Plot to file

Plot to file

Allows you to plot a large bitmap directly to a .png file without the added system requirements of keeping the whole bitmap in memory. The Target group sets the bitmap resolution (2400X1800 to 14400X10800.) Click on Okay to set the target file name and start a new plot to file. Click on Stop to stop a file plot in progress. Click on Continue to continue a file plot that was stopped (There is no continue after you exit this routine.) Click on Exit to exit this routine. A progress bar shows the amount of plotting finished and what remains to plot. Note: the 3200X2400 bitmap size is suitable for 8 1/2X11 printouts at 320-720 dpi. The larger bitmap sizes are suitable for poster-size printouts. This option is not available with solid-guessing, symmetry (except vertical symmetry) in the New Formula window or with anti-aliasing.

4.7 Edit Paste Data command

Paste Data command (Edit menu)

Use this command to paste the data in the file "c:\zcopy.zp" to the active view. The palette stored in the file is copied to palette 10(F11).

Shortcut

Keys: CTRL+R

4.8 Parameters Window

Parameters Window

There are edit controls for entering the complex constant (real and imaginary parts), and the min/max ranges for the real and imaginary window coordinates. Firefly uses three-corner plotting for easier rotating, so boxes are provided for Top Left, Top Right and Bottom Right real/imaginary coordinates. These reflect the current range values that may have been derived from zooming with the Zoom option. Edit boxes handle the number of iterations (0-65535) and the bailout (0-65535). When the bailout is 0, the convergence limit (formula window) is used as bailout reference for escape and convergent-type formulas. Cj, ck, hj and hk are for entering hypercomplex parameters. Cx,cy,cz and cw are used as complex C increments for Julia-Tower types. The Size slider controls the overall size of the picture. The Size slider sets the horizontal resolution, while the vertical resolution is then scaled according to the full-screen VGA ratio, 4 to 3(1:1 if that aspect is selected through the Auto menu.) The Sector slider controls which of 4 sectors the picture will be drawn in, if the Size is less than or equal to (the full-screen horizontal resolution)/2. Otherwise the picture is centered according to the full-screen dimensions. This allows you to show zooms of a particular function by using different sectors, or show the affect of different plotting options. Each sector is erased individually. Note: if you try to continue a plot in a different sector than you started with, the plot will continue in the original sector. The Thumbnail button next to the Size slider is used to set a thumbnail size quickly. The thumbnail size toggles between 1/4 and 1/8 of the horizontal screen resolution, e.g. 200X150 or 100X75 for an 800X600 screen.

The more iterations used, the longer it takes to plot a function, but more detail will be present. 10-20 is sufficient for most biomorphs, while more iterations will be required for Mandelbrot and Julia sets, depending on the detail required.

Start color and end color boxes are provided to limit the palette colors that the current image uses. Use start color of 0 and end color 235 to use the full palette. A split palette such as a divide-by-eight palette is easier to use when the end color is 231. The color-cycling keys (arrows and enter/backspace) work for only those colors that the start/end colors designate. Note: Firefly scales between each color index to add 255 additional colors. The actual palette used in Firefly is around 60000 colors, though not all of these may be unique.

Select the Okay button to start a new plot from column 1. Select the Continue button to continue a plot at the row it left off, if it is not a complete drawing. The screen or sector is always cleared before beginning a new plot.

The Reset button returns all boxes and slider values to their original values when the window was opened.

The Rotate box is used to rotate the picture by any degree. This rotates all three points that define the Z-plane. The rotate angle is reset to zero after each use.

The File button (when enabled) allows you to print a large bitmap (2400X1800 to 14400X10800) directly to a .png file instead of keeping the bitmap in memory. An external viewer may be necessary to view or print the larger bitmaps thus created. The File option is not available for solid-guessing and anti-aliased pictures or symmetry (except for vertical symmetry.)

4.9 Dll Configuration command

Plug-in Configuration command (Edit menu)

There are list boxes here for each type of plug-in that Firefly handles, formula, rendering and filter plug-ins. Firefly has a set of three default plug-ins that are loaded at start-up time, "fractal.fll", "fractal.rll" and "fractal.fil" (Never delete these files, without installing replacement plug-ins with the same names! These files should always be installed in the default plug-ins directory.) Any other Firefly-compatible plug-ins in the plug-ins directory are also loaded at start-up time. You choose a formula, rendering method or filter and press the add button for which "action" you want to add to the action list. The action listbox contains the list of actions that Firefly processes in its iteration loop to produce an image. Usually it takes at least one formula and a rendering method to produce an image. (Some pictures could be done with just a rendering method or two, but these are not fractals...)

The Delete button removes the displayed action from the action list.

The Change button changes the currently displayed "type" of action to another action of the same type.

Click on Okay to save changes made to the action list and redraw the current fractal image(if Image/Auto Redraw is enabled.)

Click on Cancel to exit the configuration editor discarding changes made to the action list. Note: using the close box to exit the configuration editor retains any changes made to the action list, but the image is not redrawn immediately.

Some parameters/variables are associated with each type of action. Firefly now supports v1.03 of the Ferguson/Gintz plug-in spec. This provides for plug-ins that contain their own configuration dialogs. Each action may contain its own entry method for user-defined variables, which appears when the action is added or changed, or the Configure button is clicked. A few additional entry boxes are provided for certain variables that affect all the rendering actions after the iteration loop is exited. The Slope variable acts as a palette multiplier when less than 1.0. This spreads out the colors when a rendering method sometimes doesn't fully use the palette in Firefly. Post Render and Post Fractal Dim. can be used to magnify the intensity of a rendering method. The Post Render factor is applied to the rendering pens x and y, while the Post Fractal Dim factor in applied to the fractal dimension variables used with Steven C. Ferguson's later rlls. The fractal dimension method works best with a non-palette mode, though is useable with the palette mode. Note: in palette mode, the Post Render factor is only applied in conjunction with a non-zero Post Fractal Dim factor. This allows FF to maintain compatibility with images done with previous plug-in versions. (For earlier plug-in versions, when an action is configured a default dialog is displayed that contains assorted edit boxes for variables that may or may not be applicable to the action being configured.)

The Filter Type applies in cases when there is more than one rendering method in an action list, or the list contains one or more filters. Rendering actions can be divided into two groups, active and passive. Active rendering actions usually evaluate the return values of a formula, and if they meet certain conditions, cause the iteration loop to terminate or "bailout." Passive rendering actions and filters usually return coloring info, but do not directly cause a "bailout". All rendering actions and filters return a post-rendering or post-filtering value for coloring purposes, whether or not they cause the iteration loop to bailout. When the Type is set to background, all passive rendering actions have their post-rendering values applied to the "active" post-rendering value, if it is non-zero. In this case background pixels would remain uncolored. When the type is Add Offset, the post-rendering values of passive rendering actions is added to all of the return values of the active rendering action. When the type is Bailout Only, only the return value of the active rendering action is used. For filters, the type variable has a reverse effect. For a background-type filter or Bailout only, the filter is applied to only pixels where the final post-rendering value is zero. When the filter is used with Add Offset, it is added to all pixels. The use of the background type allows images that contain filtered backgrounds, without filtering the rendered foregrounds. In many cases, such as orbit-trap pictures it is desirable to prevent a filter from adversely affecting the 3D-coloring of the stalks. The Add Offset type is useful when you want to mix different rendering methods, active and passive, such as bubble and atan, the results of which may not contain any "background" pixels..

The action variables can be different for each action, except the Filter Type, which affects all rendering and filter actions the same.

One other edit box displays the filename for each action in the action list. Tip: to minimize problems reloading images, keep the plug-ins in the default plug-ins directory. This directory may be changed through the Image/Preferences command, but you must exit the program and reload it to initialize the new plug-ins directory

4.10 Fractint Variables/Options

Edit Fractint Variables/Options

To help new users of Firefly who may have past experience with Fractint, this window contains all the variables in FF that have Fractint equivalents or near equivalents. Firefly variables such as cr and ci are relabeled with their Fractint counterparts, in this case the real and imaginary parts of p1. Rendering equivalents for inside and outside coloring, passes and bailout tests are also made available for editing. A small formula window allows one to browse forms for function parameters, etc. Note: the default selections in the Bailout Test, Outside and Inside Color list boxes maintain current (non-Fractint) options in these categories. Potential Slope turns on continuous potential when non-zero, as FF contains no Max Color for this method.

4.11 Size

Size

This allows you to set the drawing area for a picture, independent of the Windows screen size. It also shows which size is currently in use. The aspect for the drawing is based on the ratio of X(horizontal width) to Y(vertical height.) The custom setting allows for any size/aspect that system memory will permit. Videos are limited to the standard 4/3 vga aspect or 1/1.

Midi output is limited to images with the standard 4/3 aspect. The minimum size for an image is 40X30.

4.12 RGB Thresholds

RGB Thresholds

When Firefly plots pixels using a simulated palette, the colors are normally scaled (interpolated) to fit between one of 236 color indexes. The palette editor can edit these 236 colors. Sometimes you don't want an intermediate color to be plotted, if adjacent color indexes differ a lot.

There are two tests that can be made to determine the degree of color difference between adjacent color indexes. The first is RGB, where the difference in individual components is tested. The thresholds are set so that color indexes that differ in values exceeding the individual RGB thresholds will bypass the interpolation process. The values can be set from 0 to 255, with 255 the default. At 255, all pixels are interpolated except the first and last colors. At 0, no pixels are interpolated.

Test 2 measures the difference in the sum of the RGB values of adjacent colors. If the sum is greater than a preset limit (0-765), no interpolation is done.

These two tests can be combined with the Both option.

4.13 Edit Text command

Text (Edit menu)

Allows you to edit text and font and apply it to a drawing. Select the font button to set the font style, size and color. In the text window click on Okay to add a line of text to the current image. (You can add multiple lines of text too, up to 80 characters.) The cursor will change to a crosshair. Position the cursor where you want the text to start and left-click the mouse. Note: font and title text are saved in the file "prefs.txt" in Firefly's startup directory. Title text can also be edited (as a single line only) in the Edit/Formula window.

5 Image menu

Image menu commands

The Image menu offers the following commands:

Draw	Draw the picture.
Draw Composite	Draw composite from figures 1-4.
Auto Redraw	Redraw image on command.
Auto Clear	Clear drawing area before new plot.
Auto Sound Alert	Enable or turn off sound alerts.

Auto Remote	Open remote automatically at startup.
Auto Time	Show time used to plot image.
Merge Sum	Merge current pixel color with previous color summing colors.
Merge And	Merge current pixel color with previous color anding colors.
Merge Or	Merge current pixel color with previous color oring colors.
Merge High	Merge current pixel color with previous color by choosing highest rgb.
Merge Low	Merge current pixel color with previous color by choosing lowest rgb.
Merge Back	Merge current pixel color with previous color by excluding background color.
Merge Diff	Merge current pixel color with previous color by using difference of colors.
Abort	Abort drawing.
Continue	Continue drawing.
Zoom	Zoom into rectangle.
New View on Zoom	New view on zoom.
Picture	View image full-screen.
Reset->	Reset coordinates, current figure or all figures
Figure 1	Switch to figure one.
Figure 2	Switch to figure two.
Figure 3	Switch to figure three.
Figure 4	Switch to figure four.
Composite	Select figures to merge.

5.1 Image Draw command

Draw command (Image menu)

Use this command to draw or redraw the image for the current fractal variables. Clicking inside the draw window with the left-mouse button stops all plotting. Use the Continue command to restart plotting from the current column.

5.2 Image Draw Composite command

Draw Composite command (Image menu)

Use this command to draw or redraw an image defined in the Composite command as a merging of figures 1-4. Clicking inside the draw window with the left-mouse button stops all plotting. Continue is disabled for this command.

5.3 Image Redraw command

Auto Redraw command (Image menu)

With this command disabled (on by default), redraw does not occur except when the Draw command is executed, or Continue. Most of the time you want to see the results of changing a parameter or mapping option, so redraw occurs automatically with parameter or mapping changes. Sometimes you want to change more than one parameter before redrawing the image, or you may want to *continue* a mid-point displacement plot (after loading the file, this is the only way to recreate the plot.) So you need to turn this option off then.

5.4 Image Auto Clear command

Auto Clear command (Image menu)

With this command enabled (on by default), the drawing area is cleared before starting a new plot. You can turn off this option when you want to see the effect of minor changes to parameters, as they affect the plot pixel by pixel, or when setting up a multiple-layered fractal, as in a 3D landscape. You can use the shift-c command (hot keys) to clear the drawing area at any time.

5.5 Image Auto Alert command

Auto Sound Alert command (Image menu)

With this command enabled (on by default), the user is notified by a sound clip when a drawing is completed or user-canceled. By disabling this command the completion exclamation is suppressed and also any alert that contains a message box. Note: some sound clips are automatically generated by Windows, or there is no text alert for a given error condition. In these cases the sound alert is unaffected by the Auto Alert command.

5.6 Image Auto Remote command

Auto Remote command (Image menu)

With this command enabled (on by default), the remote is opened immediately at program startup. Handy if you find the remote useful and don't want to click on the toolbar button each time the program starts up.

5.7 Image Auto Time command

Auto Time command (Image menu)

With this command enabled (on by default), the time that an image takes to plot is displayed when the plot is complete. Firefly saves the condition of this option at session's end, so if you disable it and close the program, the option will be disabled when you restart Firefly.

5.8 Image Merge Sum command

Merge Sum command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using a summing algorithm. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.9 Image Merge And command

Merge And command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using an anding algorithm. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.10 Image Merge Or command

Merge Or command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using an oring algorithm. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.11 Image Merge High command

Merge High command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using the highest rgb values of both images. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.12 Image Merge Low command

Merge Low command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using the lowest rgb values of both images. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.13 Image Merge Back command

Merge Back command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using the rbg components of the new image if the new color index is not zero (for non-palette modes, rgb values of zero are excluded); else the old rgb values are retained. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.14 Image Merge Diff command

Merge Diff command (Image menu)

With this command enabled (off by default), current pixel color is not overwritten when a new image is drawn. Instead the colors are merged using the difference of the rgb values of both images. The auto-clear option must be disabled and solid-guessing off to choose this option. Useful to merge two or more separate fractal images/types with the initial image(s) "bleeding" through.

5.15 Image Abort command

Abort command (Image menu)

Use this command to stop drawing. Clicking inside a window's drawing area or close box (or the program close box) will also stop the drawing. Note: once a plot has started Firefly continues to draw the image for that window regardless of which drawing window has the input focus, until done or aborted. You can open and close other drawing windows without affecting the current drawing, but only one drawing is active at any time.

5.16 Continue Draw

Continue Draw

Continues a plot that was aborted early. The plot is restarted at the beginning of the last row drawn. Continue is disabled when an Image/Merge option is selected.

5.17 Zoom

Zoom (Image menu)

Turns on zoom mode, so that detail of the current plot may be magnified. Alternatively, just click inside any drawing window, move the mouse, and the zoom box will appear. Using the mouse, move the zoom box over the portion of the plot you wish to magnify. Hold the left mouse button to shrink the box or the right button to enlarge it. Use the left and right arrow keys to rotate the box counter-clockwise or clockwise.

New: alternatives for shrinking and enlarging the zoom box make use of the keyboard keys Shift and Ctrl. Press Shift to enlarge the box X4, or Ctrl to shrink the box to 1/4. Press the Insert key to rotate the box 45 degrees, or the Delete key to rotate -45 degrees.

Use the up and down arrow keys to squash or expand the box, changing the aspect of the image. You start a zoom by pressing the space bar. You abort a zoom by clicking outside the main window or in the title bar, or by pressing the escape key. The program will begin a new plot at the new coordinates. You may zoom in by defining a box inside the current drawing area. You zoom out by drawing a box outside the current drawing area. The outer zoom limits are between -1000 and 1000. The precision is that of double precision (64 bits)
Note: Zooming in a three-dimensional plot is not supported, nor is zooming on a spherical or

random (midpoint displacement) fractal. If you change screen resolutions, you must redraw the bitmap image for a function before you can accurately zoom on it.

5.18 Image New View on Zoom command

New view on zoom (Image menu)

With this option enabled, a new window is opened with each zoom, instead of the zoom box area replacing the original image. Ignored in avi mode.

5.19 Picture

Picture

Displays the entire plot. expanding or shrinking the image to fit in a maximized window without title bar, scroll bars or menu bar. At all other times, part of the picture is hidden by the inclusion of the title bar, toolbar, scroll bars and menu bar. To exit full-screen mode, press any key or click the left-mouse button.

5.20 Reset

Reset

Reset the current figure or all figures to an empty Mandelbrot. All functions in the New Formula data are blanked. All options on the Flags menu are reset to their default settings. The Print Function Data ignores any reset figures.

The Ranges Only command resets only the real Z and imaginary Z ranges in the Parameters window (to +/-2.0 and +/-1.5.) No other menus or variables are affected. This is useful in conjunction with the "P" command to generate and view Julia sets. After setting the complex-C variable via shift-P (Caps Lock off), you need to reset the Z ranges to see the entire Julia set after zooming into a Mandelbrot set. The Reset All option resets all figures and clears the action list too.

5.21 Figure #1

Figure #1

Switch to Function #1. Current settings are saved under the previous image. Dll action lists are only used with Figure #1.

5.22 Figure #2

Figure #2

Switch to Function #2. Current settings are saved under the previous image. Dll action lists are only used with Figure #1.

5.23 Figure #3

Figure #3
Switch to Function #3. Current settings are saved under the previous image. Dll action lists are only used with Figure #1.

5.24 Figure #4

Figure #4
Switch to Function #4. Current settings are saved under the previous image. Dll action lists are only used with Figure #1.

5.25 Image Composite command

Composite command (Image menu)

Opens the Composite Figure window, where you can define a set of figures to merge into one image. All the merging options in the Merge Color menu are supported, plus "ALL" which is usually used for the first figure to be drawn. The "ALL" option transfers all rgb information for a figure to the drawing area, without checking the rgb state of the pixel. You can define up to four figures (layers), as part of the composite, but each figure should contain an image (if used in the composite.)

6 Type menu

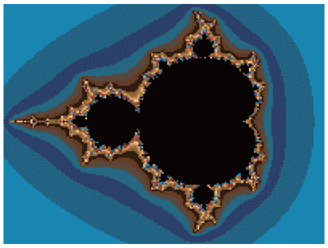
Type menu commands

The Type menu offers the following commands:

Mandelbrot	Mandelbrot set.
MandelbrotP	Mandelbrot set (orbit starts at pixel.)
Julia	Julia set.
Julia Tower	Julia tower of varying C.
Zero Init	Sets initial 'z' to zero for Mandelbrot sets.
M-Set Method	Draw M-Set fractal type.
N-Set Method	Draw N-Set fractal type.
T-Set Method	Draw T-Set fractal type.

6.1 Mandelbrot

Mandelbrot
Mandelbrots base their mapping on varying inputs of complex C, which corresponds to the min/max values set in the Parameters window. With Mandelbrot0, Cr and Ci represent the initial value of Z before the first iteration. This is normally zero, but can be changed to produce non-symmetrical Mandelbrots, or Mandelbrots based on formulas whose initial value of Z must be non-zero to generate anything.



Mandelbrot set

6.2 MandelbrotP

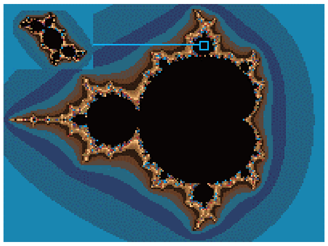
MandelbrotP

Mandelbrots base their mapping on varying inputs of complex C, which corresponds to the min/max values set in the Parameters window. With MandelbrotP, the initial value of Z is set to the value of the pixel being iterated. This produces interesting effects with some Mandelbrot formulas that normally start their orbits at zero.

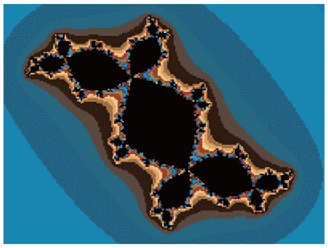
6.3 Julia

Julia

Julia sets normally have a fixed complex C, with varying inputs of Z, which corresponds to the min/max values set in the Parameters window. This option, without the Bound flag set, generates the so-called 'filled-in' Julia set, which includes non-escaping points as well as the Julia set.



Julia from Mandelbrot

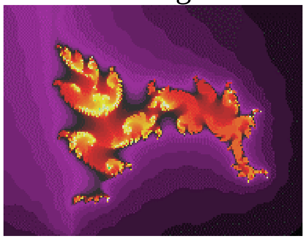


Julia set

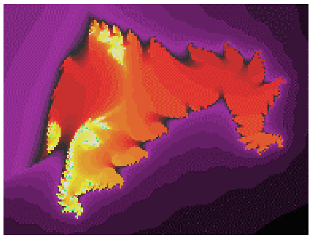
6.4 Julia Tower

Julia Tower

Julia sets normally have a fixed complex C, with varying inputs of Z, which corresponds to the min/max values set in the Parameters window. The Tower option, without the Bound flag set, generates a "stacked" Julia set. This Julia set has its constant incremented by a value of Cx, where Cx represents four variables (cx, cy, cz and cw) for incrementing each part of the complex C (cr,ci,cj,ck), specified in the Parameters window. The constant is scaled from its initial value c to the value of c+Cx. Each row of the set has a different slice of the constant. Some functions that rely heavily on both cr and ci for their shape show a marked "meltdown" as a tower. Note: zooming with this flag set changes both complex C and Cx to maintain the correct scaling factors.



Original Julia set



Julia Tower

6.5 Type Zero Init command

Zero Init (Type menu)

Use this command to set initial z to zero, excluding the complex constant, before iterating each pixel. Used with Mandelbrot types only. Normally z is set to complex c before iterating Mandelbrot sets. (The complex c may be non-zero to "warp" the orbit.)

6.6 Draw MSET command

Draw M-Set method command (Type menu)

Use this command to switch to M-Set method. Escape routines are included in the iteration loop, independent of normal dll processing, to take effect according to parameters suitable for Mandelbrot plots. Only effective when plug-in options are used exclusively to draw the image. If any built-in formulas or rendering options are used (such as decomposition or level curve), this option is ignored.

6.7 Draw NSET command

Draw N-Set method command (Type menu)

Use this command to switch to N-Set method. Escape routines are included in the iteration loop, independent of normal dll processing, to take effect according to parameters suitable for Newton (convergent) plots. Only effective when plug-in options are used exclusively to draw the image. If any built-in formulas or rendering options are used (such as decomposition or level curve), this option is ignored.

6.8 Draw TSET command

Draw T-Set command (Type menu)

Use this command to switch to T-Set method. No escape routines are included in the iteration loop beside what the plug-in options provide (Actually, a fail-safe bailout is always included, for out of range conditions.) Only effective when plug-in options are used exclusively to draw the image. If any built-in formulas or rendering options are used (such as decomposition or level curve), this option is ignored.

7 Map menu

Map menu commands

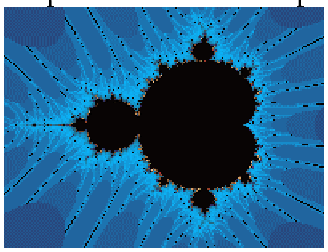
The Map menu offers the following commands:

Z-Real	Mapping based on real part of z only.
Z-Imag	Mapping based on imaginary part of z only.
Abs(Z-Real)	Mapping based on absolute value of real part of z.
Abs(Z-Imag)	Mapping based on absolute value of imaginary part of z.
Z-Real + Z-Imag	Mapping based on sum of parts of z.
Abs(Z-Real)+Abs(Z-Imag)	Mapping based on absolute value of parts of z.
>Abs(Z-Real) or Abs(Z-Imag)	Mapping based on highest absolute value of parts of z.
<Abs(Z-Real) or Abs(Z-Imag	Mapping based on lowest absolute value of parts of z.
Abs(Z)	Mapping based on absolute value of z.

7.1 Z-Real

Z-Real

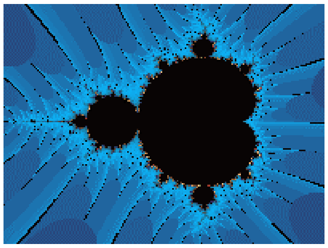
Map based on the real part of the complex number Z; used to map exponential Julia sets, etc.



7.2 Z-Imag

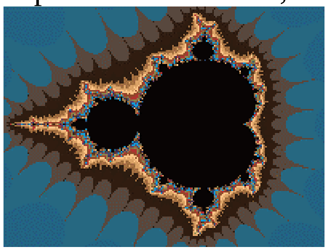
Z-Imag

Map based on the imaginary part of the complex number Z.



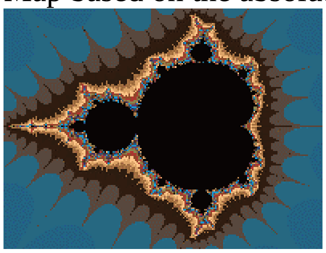
7.3 Abs(Z-Real)

Abs(Z-Real)
Map based on the absolute value of the real part of the complex number Z; used to map exponential Julia sets, etc.



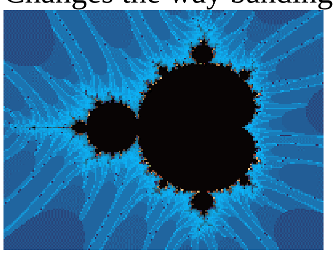
7.4 Abs(Z-Imag)

Abs(Z-Imag)
Map based on the absolute value of the imaginary part of the complex number Z.



7.5 Z-Real+Z-Imag

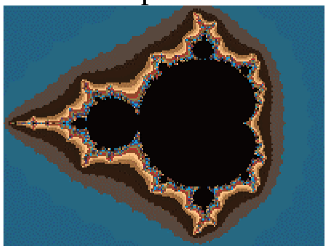
Z-Real + Z-Imag
Map based on the sum of the real part and the imaginary part of the complex number Z. Changes the way banding appears in complex mappings.



7.6 Abs(Z-Real)+Abs(Z-Imag)

Abs(Z-Real) + Abs(Z-Imag)

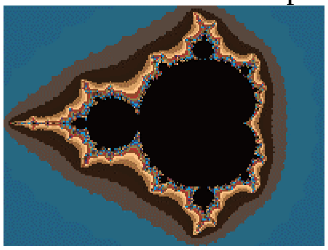
Map based on the absolute value of the real part plus the absolute value of the imaginary part of the complex number Z. Changes the way banding appears in complex mappings.



7.7 >Abs(Z-Real) or Abs(Z-Imag)

>Abs(Z-Real) or Abs(Z-Imag)

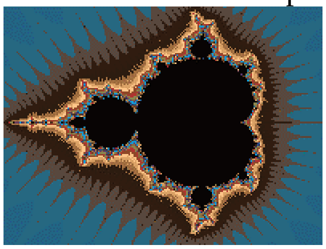
Map based on the greater of the absolute value of the real part or the imaginary part of the complex number Z. Works like a logical 'or', where either part of z must exceed bailout to break the iteration loop. Changes the way banding appears in complex mappings.



7.8 <Abs(Z-Real) or Abs(Z-Imag)

<Abs(Z-Real) or Abs(Z-Imag)

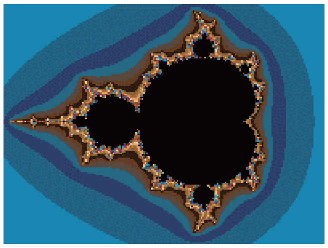
Map based on the lesser of the absolute value of the real part or the imaginary part of the complex number Z. Works like a logical 'and', where both parts of z must exceed bailout to break the iteration loop. Changes the way banding appears in complex mappings.



7.9 Abs(Z)

Abs(Z)

Map based on the absolute value of the complex number Z (traditionally calculated by taking the square root of the sum of the squares of the real and imaginary parts of Z, but Firefly uses only the 'sum'(modulus of z) for break-point tests.) The standard method of mapping Julia and Mandelbrot sets.



8 Break menu

Break menu commands

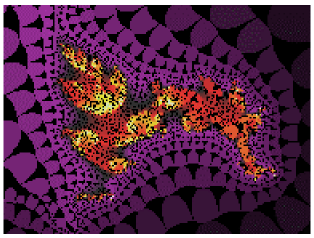
The Break menu offers the following commands:

Biomorph	Escape on any part of z.
Bioconvergence	Relates Biomorph method to convergence.
Biomorph Off	Reset biomorph flag.
Orbit Traps	Set orbit trapping method.
Newton->	Define Newton method or clear Newton flag.
Renormalization	Renormalization.
Convergence	Color converging points with level set colors.
Period Check	Periodic checking (speed option).
Default Function	Use default function type.

8.1 Biomorph

Biomorph

Biomorphs test the real Z and imaginary Z values after breaking the iteration loop. If the absolute value of either is less than the preset bailout, the point is mapped as part of the set. This method produces biological-like structures in the complex plane. Normally the biomorph tendrils are colored in the set color(the color reserved for non-divergent or inner points.) With the Set Only flag on, the tendrils are colored according to the color-scaling option used(other external points are colored in the background color.) A window is opened each time this option is selected to set a color for the area that falls within the biomorph trap. This can be 0-235.



8.2 Bioconvergence

Bioconvergence

This option relates the Biomorph method to convergence for convergent-type fractals(Newton, Renormalization and Convergence.) A pseudo-biomorphic algorithm is applied to converging points.

8.3 Biomorph Off

Biomorph Off

Turns off the biomorph flag, including bioconvergence too. Alternatively you can enter -1 in the Biomorph window to turn off the bio-flag.

8.4 Orbit traps

Orbit Traps

This includes methods that trap the orbit of a point if it comes in range of a pre-specified area or areas.

The Epsilon-Cross method colors points only if the absolute value of Z-real or Z-imaginary is less than or equal to Epsilon (a small value.) Other points are mapped at the time they blow up (exceed the bailout.) This produces hair-like structures that branch wildly from the complex set boundaries. For the Epsilon-Inside option, the epsilon method is applied only to points included in the set. For the Epsilon-Outside option, the epsilon method is applied only to points outside the set.

The Globe method uses a circular area around the origin to map a point's orbits. This produces sphere-like structures.

The Ring method uses an area formed by two circles around the origin to map a point's orbits. This produces ring-like structures.

The Four-Circles method (Paul Carlson) uses four circular areas to map a point's orbit. This produces sphere-like structures.

The Square method uses an area formed by two squares around the origin to map a point's orbits. This produces ring-like structures with right angles.

The Formula option allows you to enter your own formula for an orbit trap in the Formula box in the Edit Formula window. This works for built-in formulas and fractal types except type 9(formula) and 5(random.) An example of how to specify an orbit trap is the following formula for the ring method: a=x#*x#,b=y#*y#,x#=a+b-.25,x#=abs(x#)
where x# is the real part of z and y# is the imaginary part of z at the nth iteration. X# is then compared to the epsilon and epsilon2 values. If x# is less than epsilon and greater than or equal to epsilon2 then x# is subtracted from epsilon and the resulting value is used for coloring purposes(a level curve must be chosen as an option.) This is also the loop-breaking condition.

Epsilon2 is used to create windows into the stalks. The default value is 0.0, which produces solid stalks.

The Parametric Formula option allows you to enter your own parametric formula for an orbit trap in the Formula box in the Edit Formula window. This works for all built-in formulas and fractal types except type 9(formula) and 5(random.) An example of how to specify this type of orbit trap is the following parametric formula (used for Paul Carlson's fast mset picture): z=0.3*(cos(pixel)^3+i*sin(pixel)^3).

Here, pixel is used to specify the polar angle of z(instead of its usual c or z-plane value), where pixel = atan(imag(z)/real(z)). Note that z is used for setting up the distance variable instead of x# in this case.

The Display Even Only option is used to de-clutter some epsilon plots by coloring points that escape on even iterations only. Odd points are plotted in the background color.

A window is opened to enter a value for Epsilon and Epsilon2, which are used to define the size of the trap areas (.001-2.0 and 0.0-epsilon.) The exclude box is used to exclude the first # iterations (0-99) from orbit trapping.

To produce the maximum 3-D effects (as Phil Pickard and Paul Carlson do) with these options, Level Curve #4 must be set, and the Cutoff value (in the Parameters window) should equal the negation of the epsilon value (-epsilon.) You'll need to set up a special palette with a number of color ranges that matches the split-palette number if set. Built-in examples d3-d6 illustrate how to set up 3d-like fractals.

To automate the process of producing Paul Carlson's 3-D like fractals, a check box has been added to this window for 'Carlson extensions'. This sets the Background and Set Only flags as well as the Level Curve #4 and the cutoff value, and sets the Map to <Abs(Z-Real) or Abs(Z-Imag). An exclude value of 2 is also used. The Map and exclude values are extra parameters that Paul uses in some of his formulas, and may be omitted in other cases. This is easily done by first enabling the Carlson extensions by checking the box and clicking on Okay, then opening the window again and un-checking the box and changing the appropriate variables/flags.

8.5 Newton Set

Newton Set

The Newton flag is used to map the zeros of a particular function after the Newton transformation has been applied to the function. The program doesn't make the transformation (z-(f(z)/f'(z)), where ' stands for d/dx), but it does allow you to map up to 6 attractors. Each time the Newton flag is set, a window is opened to allow you to enter up to 6 attractors (or repellers) of the function.

This flag is mutually exclusive with the Boundary Scan, Convergence and Renormalization flags, and automatically excludes all points that don't converge to one of the attractors set, within the preset number of iterations. The points that converge are colored with one of up-to-6 possible color spreads(the built-in functions may allow more colors) evenly-spaced in the current palette, according to the root they converge to and the time it takes to converge. The non-converging points are mapped with the set color or their level set color(with a level flag set) after the maximum number of iterations. Non-converging points show up typically as round areas or spots.

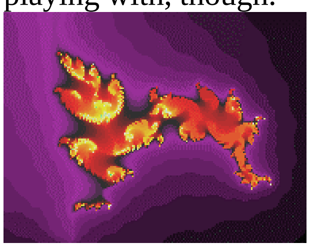
Generally, a limit of 50 iterations gives optimum results. The Newton transformation is normally used with Julia sets, as the attractors (solutions of the formula) can be calculated beforehand. Its also possible to explore the Mandelbrot set applied to Newton's method, but only with some of the built-in formulas mentioned above. It this case, the solutions of the formula for every point on the screen have to be calculated separately, which the program does in a dedicated routine.

8.6 Newton Off

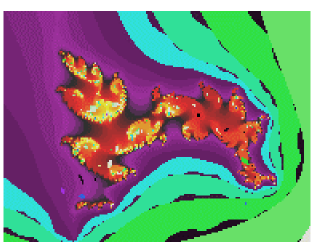
Newton Off
Turns off the Newton flag, otherwise this option is disabled.

8.7 Renormalize

Renormalization
The Renormalization flag uses a hierarchical lattice transformation to map magnetic phases, with either the Julia set or Mandelbrot set as the iterated function. (Consult The Beauty of Fractals by Pietgen and Richter for appropriate formulas to use.) Basically, the default-mapping algorithm checks orbits for convergence to 1 or infinity, and scales these points in different colors. This flag is mutually exclusive with the Newton, Convergence and Boundary Scan flags. The default method of display actually only checks if z passes through 1. This is similar to the epsilon-cross mapping method. For the original renormalization formulas, there is a strong orbital attraction to 1. For other functions, this mapping produces unusual effects(with obscure mathematical foundations.) For the built-in functions, the convergence tests (type 2 or 3 and 6-8 display types) are the same as the ones used with the Newton flag. So either method produces similar results with the same formula. The differences are worth playing with, though.



Original picture



With renormalization

8.8 Convergence

Convergence
With the Convergence flag set, the program does a convergence/periodic check on all points. This is similar to the convergence checks done with Newton and Renormalization, but also the orbits of each point are saved to determine if the orbit repeats. When an orbit repeats, the iteration loop is broken and the point colored according to its break time. Depending on the iteration limit, the last 200 points of each orbit are tracked for this check. This flag is mutually exclusive with the Newton, Renormalization, and Boundary Scan flags. May use Newton display methods 3 and 6-8(alternate convergence tests), by setting the arg gadget to these values.

8.9 Period Check

Period Check

With the Period Check flag set, the program does a convergence/periodic check on all points. This is similar to the convergence checks done with the Convergence option, except that only the orbit is escaped from when it repeats. The color of the pixel at escape time is the set color, when no other inside-coloring methods are used. With boundary scan on, the pixel is set to the background color. With continuous potential or level-mapped curves (color menu options), the pixel color will be altered later. This option is useful for speeding-up plots with high iterations, and a large attractive(non-escaping) area. There is a chance of interaction with some inside-mapping options, such as Level Curves, so use with caution.

This flag is mutually exclusive with the Newton and Renormalization flags. May use Newton display methods 3 and 6-8(alternate convergence tests), by setting the arg gadget to these values.

8.10 Default Function

Default Function

When this option is enabled (off by default), convergent functions are iterated according to their original type. Firefly allows treating a renormalization curve as a Newton curve, or vice versa, but the governing flag must be set through the flags menu. The Default Function option allows a built-in function to work as a Newton or renormalization curve without those flags being set. Newton functions work only as Newtons and likewise for renormalization formulas. Function types that use two built-in functions can distinguish between convergent and non-convergent formulas and use the suitable escape or convergent checking for each formula.

9 Render menu

Render menu commands

The Render menu offers the following commands:

Boundary Scan	Boundary-scanning method.
Level Curve->	Set level curve or reset level curve flag.
Decomposition->	Binary or continuous decomposition.
Decomposition Off	Reset decomposition flag.
Switch->	Switch z components or z for c.
Spin	Increment C by scaled factor of cx at every iteration.
Filter	Choose an optional tail-end filter.
Anti-Alias	Use anti-aliasing, with 1X4 or 1X2 super-sampling.

9.1 Boundary-Scan

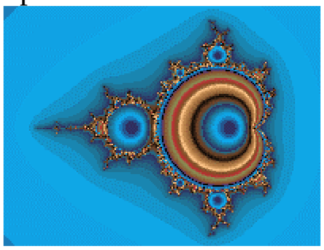
Boundary Scan

This option generates complex sets using a boundary-scanning routine described by C. Pickover. This flag is mutually exclusive with the Convergence, Newton and Renormalization flags.

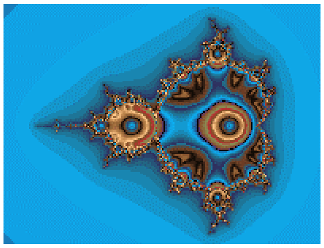
9.2 Level Curve

Level Curve

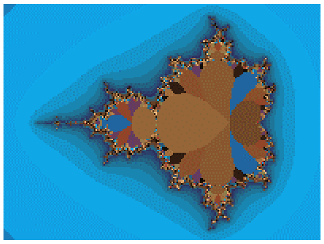
Level-curves map the set points based on how small the value of Z gets. This allows the inside of the complex set to be color-scaled. Log Map #1 produces colored bands on the inside of the complex set. Points are mapped according to what the value of z is at final iteration. Small Log #2 and Linear Map #5 produce circular patterns inside the complex set. Points are mapped according to the smallest value z gets during iteration. Indexed Log #3 and Indexed Linear #6 are mapped according to the time it takes z to reach its smallest value. Level curves 2,3(and 5,6) are described more fully in The Beauty of Fractals. Linear Map #4 is mapped like Log Map #1 (with the mapped value of the function at its final iteration applied to the color palette) and produces 3D-like effects with the Epsilon-Cross method. The Log methods use a log palette, while method #4-6 use linear palettes. This option can override (or may be overridden by) many of the options in the Color-Scaling menu. Decomposition doesn't use Level Curve shading, unless you select the Use Level Curve option.



Log Map #1



Small Log #2



Indexed Log #3

Bubble #7 uses Paul Carlson's contour-mapping method to produce 3D-like bubble pictures. The method is very sensitive to which formula is used, working best with the basic Mandelbrot set z^2+c and the like. Color-mapping should be set to Use Level Curve. This is a trial and error method that uses two other variables to produce the final effect, magnify and cutoff, as entered in the Parameters window. Magnify is used to screen unwanted background contours in the plot, while cutoff is used to fill out the color palette. Magnify should be a low value, usually less than .1, to eliminate the contours that usually appear in escape-type Mandelbrot/Julia sets. If it is too large the bubbles will be too crowded, while too small a value will cause the bubbles to disappear. Cutoff needs to be a small negative value, usually equal to the magnify value times the number of color splits. E.G., for a magnify value of .1 you should use a cutoff value of -.8 for a divide-by-eight palette. An incorrect cutoff value will cause the colors to overlap in the bubbles. For split palette pictures, the colors are divided according to their level index, as in Indexed Linear #6. The color ranges should be graded from light to dark to highlight the bubble centers.

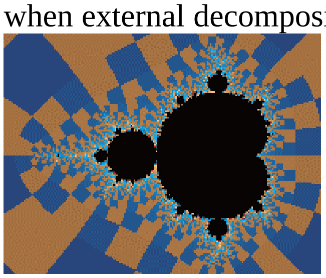
9.3 Decomposition

Decomposition

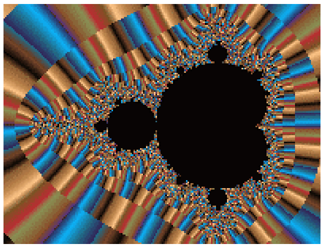
When a Decomposition flag is set, you have the option of performing either a binary or continuous decomposition. Toggle the External/Internal option for either an external or internal decomposition. The Angle-Only option excludes the Coloring-scaling options from consideration when plotting points derived from decomposition. It also excludes escape times in connection with the Angle-Iteration option on the Color-Scaling menu. An external decomposition decomposes points that are outside the complex set. An internal decomposition decomposes the complex set. For Mandelbrot/Julia curves, z-arg is broken into two parts for a binary decomposition. For Newton/Renormalization curves, the binary decomposition is also related to the number of solutions a formula has, if it supports mapping option 1. Continuous decomposition breaks z-arg into n parts, where n=angles(2-256), as set in the Continuous Decomposition window. An angles value of less than 2 turns off the Continuous Decomposition option.

Note: With the graded-palette option checked, the decomposition option is extended for extra smoothness in Firefly. The number of angles is internally multiplied by 236 to track the decomposition angle more closely.

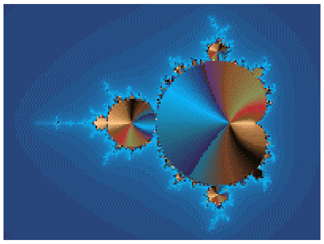
(Consult The Beauty of Fractals by Peitgen & Richter for a mathematical explanation of decomposition.) When Biomorph or Epsilon are decomposed, the tendrils or hairs are decomposed as external points. Use the Set Only flag to emphasize the tendrils and hairs when external decomposition is used.)



Binary Decomposition



Continuous External



Continuous Internal

9.4 Decomposition Off

Decomposition Off

Turns off all decomposition flags and resets the Internal/Eternal option to Eternal.

9.5 Switch

Switch

When a Switch flag is set, you have the option of switching the real and imaginary parts of Z, or switching Z for C. The real part of Z is exchanged with the imaginary part of Z after each iteration. Using this technique with the Mandelbrot set produces a tricorn-like plot. When Z is switched for C, normally you get Mandelbrots from Julia sets and vice versa. This option also offers an inverse to the Julia Tower, with glimpses into the elusive Mandel Tower, a sort of gateway to infinity.

9.6 Spin

Spin

When the Spin flag is set, the complex constant is incremented by a scaled factor of cx (cx*c/iterations) at every step of iteration. Unavailable for Julia Tower types.

9.7 Filter

Filter

Based on Stephen C. Ferguson's filter algorithms in his program Iterations, this option allows you to choose one of 29 tail-end filters to apply to any 2D plot. The name of the filter corresponds roughly to its effect on the basic Mandelbrot-squared set. The effect will vary with the formula and fractal type chosen. This overrides the Background option on the Color-Scaling menu. Useful to add detail to orbit-trap pictures, as well as perk up any otherwise ordinary picture. Filters 27-29 are generalized filters that use fn4 and fn3(in the edit/formula window) for expanded scope. The "cross" filters use the epsilon variable from the orbit-trap window. (To set epsilon's value without applying an orbit-trap, first change the value of epsilon to the value desired, and set the orbit-trap flag. Then turn off the orbit-trap flag. Epsilon value remains unchanged when the orbit-trap flag is turned off.)

The Magnify variable is used to intensify or de-intensify the effect of the filter. This value can range from 1-500 nominally. The Add Offset box is checked when you want the filter to add an offset to the color value normally plotted. The Exclude Background box works like the Add Offset box, except that background pixels are unfiltered. With the Replace All box checked, the filter totally replaces the normal color value, which can lead to very different color rendering. With the Background Only box checked, only pixels which would normally be colored with the background color (index 0) are filtered.

9.8 Anti-Alias

Anti-Alias

Applies a 2 to 1 or 4 to 1 averaging filter to every pixel plotted, to reduce jaggies and other high-frequency noise. This increases the processing time 4 to 8 times, so is mainly a final rendering method, not for general development use. Not available for 3D-type fractals (including quaternions), plot-to-file, midpoint-displacement, 3D backgrounds or with the tesseral solid-guessing method. Note: because of the lengthy time required for applying the anti-aliasing filter, and because anti-aliasing calculates different smoothing colors each time the palette is changed, all color-cycling and palette-switching hot keys are disabled with the anti-alias flag set.

10 Pixel menu

Pixel menu commands

The Pixel menu offers the following commands:

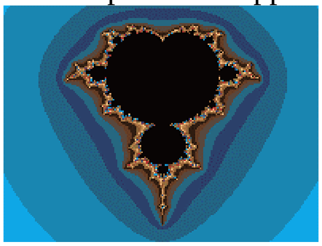
Phoenix	Phoenix Curve.
Invert	Invert image around circle.
Invert Off	Reset inversion flag.
Symmetry->	Horizontal, vertical or XY symmetry.
Fast->	Speed up five basic types of fractals.
Solid-Guessing	Solid-guessing plotting mode.
Tesseral	Tesseral solid-guessing plotting mode.
Segment	Plot image as part of segmented array.
Cliff's Slice	Use C. Pickover's 'CPCB' slice for hypercomplex planes.
Torus	Use torus method.
Torus Off	Reset torus flag.

Use Stencil Use border on picture.

10.1 Phoenix

Phoenix

The Phoenix flag rotates the planes, so that the imaginary plane is mapped horizontally and the real plane is mapped vertically.



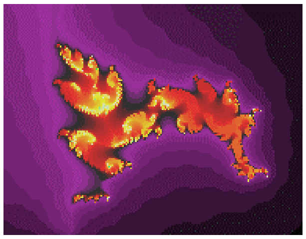
This option is normally used for mapping Phoenix curves (Shigehiro Ushiki), which are Julia-related curves based on the formula $f(z+1)=z^2+p+qz$. 'p' and 'q' are constants, and the 'z' term of 'qz' is actually the value of z^n-1, or the previous value of z before the current iteration. 'zn' is reserved by Firefly to represent this value, while the complex constant set in the Parameters window becomes 'p' and 'q'. The real part of the complex constant is 'p' and the imaginary part of the constant is 'q' (when the Phoenix option is chosen).

If the Phoenix flag is used with the Mandelbrot option, 'j' and 'k' should be used as the constants, since the complex constants p and q are already used as the starting value of 'z0'.

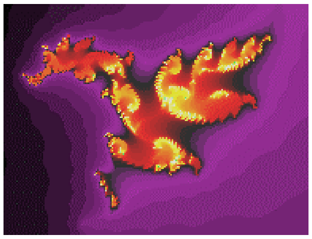
10.2 Invert

Invert

The Invert flag inverts the plane around a circle. A window is opened that allows the user to specify the circle's radius and center coordinates. Select Auto Coords to let Firefly calculate the center coordinates and circle radius. Using Auto Coords, the new radius and center coordinates are calculated when the picture is next drawn. You can zoom on an inverted picture as long as radius and center coordinates remain the same. Use the Perspective box to alter the X/Y symmetry of the inversion. A smaller Perspective value (less than 1.0) stretches the inversion in the vertical direction.



Original picture



Inverted

10.3 Invert Off

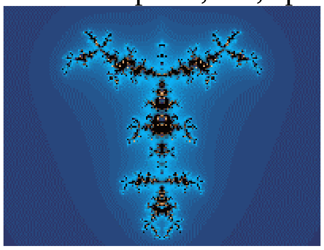
Invert Off

Turns off the inversion flag. Alternatively you can set the inversion radius to 0.0 to turn off inversion.

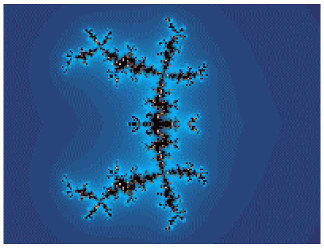
10.4 Symmetry

Symmetry

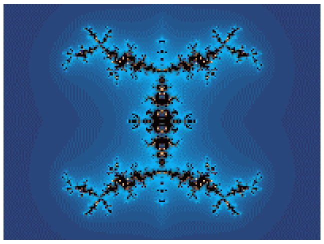
This produces a mirror image from left to right (vertical) or top to bottom (horizontal) or both (xy). You can zoom with symmetry, but the results will be uncertain if the zoom box is off-center on the window or if rotation is used. Symmetry has no effect when used with the Tesseral option, 3D, spherical or random fractals.



Vertical symmetry



Horizontal symmetry



XY symmetry

10.5 Fast

Fast

This option alters flags and variables to speed up drawing time for 5 basic fractal types.

Fast Mandelbrot/Julia sets up the flags and filters to do non-convergent type plots. This is fully optimized for the non-hypercomplex Mandelbrot set (p0; z^2+c.) New code has been included in v1.16+ to speed up the basic Mandelbrot/Julia set even more. This is based on an assembly language routine by Damien M. Jones. It's over twice as fast when used with pictures that don't need filtering or exponential smoothing.

Fast Orbit-Trap sets up the flags and filters to do a Carlson-style orbit-trap picture.

Fast Newton sets up the flags and filters to do a basic Newton's method picture.

Fast Biomorph sets up the flags and filters to do a Pickover-style biomorph picture.

Fast modes are not available with Formula Types other than 0 or 9(for Mandelbrot/Julia sets only.), or a user-defined bailout or initialization.
The speedup averages 1/3X faster on general types, but can be up to 10X on the Mandelbrot option.

10.6 Solid Guessing

Solid Guessing

In the solid-guessing plotting mode, the program guesses at colors that lie inside rectangular areas of the plot. It first computes all the perimeter pixels of a rectangle, and checks if all the pixels have the same color. If so, all the pixels inside the rectangle are colored the same and no further calculations are done on that rectangle. Otherwise the rectangle is broken into four parts and the above procedure is repeated for each part. If any of the perimeter pixels are different at this point, all the remaining pixels in the smaller rectangle are computed. The screen is updated in groups of 16 lines.

This method can be much faster than the default single-pass mode that Firefly starts in. This is especially true for plots that have large areas of a single color. For very intricate plots that have little open space, the solid-guessing mode can still be 15-20% faster. Solid-Guessing has only a slight affect on images that use the normal or generalized coloring modes.

The solid-guessing mode can fail for some plots that have small areas of one color that are islands inside the rectangular areas under test.

10.7 Tesseral

Tesseral

Tesseral is a variation of the solid-guessing plotting mode, where the program guesses at colors that lie inside rectangular areas of the plot. The main difference is in the size of the rectangle that tesseral uses to start with and how the screen is updated. Tesseral starts with the whole screen and divides that into quarters, eighths etc until it reaches a solid block or a minimum size to fill in pixel by pixel. This is a recursive routine, so the whole screen is updated every 100 recursions, or when done, rather than by lines.

This method can be much faster than the default single-pass mode that Firefly starts in. It runs about the same speed as the solid-guessing mode. The main advantage Tesseral has over solid-guessing is when there are very large areas of one color in the plot (such as a Mandelbrot island) that take a long time to compute, as when the iterations variable is set to a large number.

Like solid-guessing, Tesseral can fail for some plots that have small areas of one color that are islands inside the rectangular areas under test.

When selected, tesseral opens a window to set an optional fill color. The areas filled in by the tesseral routine are filled using this color, if zero or greater is entered. Tesseral is turned off by selecting Solid Guessing or entering a negative number less than -1 for the fill color.

10.8 Segment

Segment...

This option allows you to break a large bitmap (larger than would fit into memory) into tiles that can be reassembled later with a program such as Richard Paasen's Image Arithmetic. The Segments box defines how many tiles the image is broken into (1-225.) The number of tiles horizontally must equal the number of tiles vertically; for example, a 4X4 tiling would consist of 16 tiles total. Segments are numbered 0-(no. of segments - 1), with 0 being the upper-left corner tile and following left to right in rows to the last tile at the lower-right corner.

10.9 Cliff's Slice

Cliff's Slice

With this option set, the 4D z-planes are rotated to match C. Pickover's quaternion examples in "Computers, Pattern, Chaos and Beauty Only affects quaternion and hypercomplex plots.

10.10 Torus

Torus

Pixels are mapped around a torus, and then expanded to fit the drawing area. A generalized form of Earl Hinrichs' torus method, variables are provided for center x and center y to define the c and z radii and may both equal 0.0. Results will vary with the formula used, but resembles the warping effect found in hypercomplex images. Two versions of this method are provided: the Pixel method which uses pixel values to map the torus to the fractal space, and the Two-Pi method which uses an initial rectangle 2 pi by 2 pi to map the torus to a fractal image. With the Two-Pi method, when you zoom the rectangle's size and starting points are changed to match the zooming area. The rectangle's coordinates are saved with the fractal. If you turn off the torus flag after zooming and then reinitialize the torus flag, the rectangle reverts to a 2X2 area, so the image will change accordingly. Rotating is not supported for the Two-Pi method, but does work in a limited way with the Pixel method.

10.11 Torus Off

Torus Off

Turns off the torus flag. Alternatively you can enter a negative value to turn off this flag.

10.12 Use Stencil

Use Stencil

A border is created around the plot as it is drawn. This can be a circular or oval border. The border uses the background color. Not available with 3D plots, quaternion plots, midpoint displacement or spherical plots.

11 Color menu

Color menu commands

The Color menu offers the following commands:

Cycle	Cycle colors.
Drawing Mode	Set drawing mode for pens.
Regular Color Mode	Regular color mode (two-dimensional coloring).
I See Da Light Mode	I see da light color mode (two-dimensional coloring).
Blend menu	

The following commands can be used in Regular or I See Da Light Mode (via the 2D Coloring-Options menu):

Color Parameters	Color controls.
Color Separation	Separate red, green and blue components.
RGB	RGB color mapping.
RBG	RBG color mapping.
GRB	GRB color mapping.
GBR	GBR color mapping.
BRG	BRG color mapping.
BGR	BGR color mapping.
Triangle Algorithm	Handle color overflow by triangle algorithm.
Sine Algorithm	Handle color overflow by sine algorithm.
Gray Scale	Gray scale mapping.
Invert	Invert colors.

Palette Color Mode Palette color mode (linear coloring).

The following commands can be used in Palette Color Mode (via the Linear Coloring-Options menu):

Palette Editor	Edit palette.
Triangle Algorithm	Handle color overflow by triangle algorithm.
Divide By One Palette	No split palette.
Divide By Two Palette	Split palette into two sections.
Divide By Four Palette	Split palette into four sections.
Divide By Eight Palette	Split palette into eight sections.

11.1 Color Cycle command

Cycle command (Color menu)

Use this command to cycle colors when not plotting. Works with any coloring mode, but not with hsv filtering or anti-aliasing. Undoing an action disables the cycle command until the image is redrawn.

11.2 Drawing Mode

Drawing Mode

These options determine how the pen colors are applied to the screen, when a plug-in rendering action is used. Plug-ins may use up to three pens, X,Y and Z. The Z pen is only effective in palette-coloring mode.

11.3 Color Parameters command

Color Parameters command (Color menu)

Use this command to adjust the color controls when in regular or I see da light coloring mode.

11.4 Color Separate command

Color Separation command (Color menu)

Use this command to separate colors into red/green/blue components, if in one of the 2D coloring modes (regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.5 Color Regular mode command

Regular Color Mode command (Color menu)

Use this command to switch to Steve Ferguson's regular coloring mode. Images are colored via the selection in the Blend menu and color-controls dialog.

11.6 Color I see da light mode command

I See Da Light Mode command (Color menu)

Use this command to switch to Steve Ferguson's I see da light mode. Images are colored via the selection in the Blend menu and color-controls dialog.

11.7 Blend menu

Blend menu commands

The Blend menu offers the following commands:

linear scale	linear scale color blending.
average (x + y)/2	average color blending.
subtractive (x - y)	subtractive color blending.
sum of squares 1	sum #1 color blending.
sum of squares 2	sum #2 color blending.
sin(x*y*100)*100	sin color blending.
atan(x/y)*50	atan #1 color blending.
(x + y)/2	additive color blending.
log(fabs(1000/ Data->ytot))*700	log color blending.
atan((x*x)/(y*y))	atan #2 color blending.
atan((zx*zx)/zy*zy)	atan #3 color blending.
sin((zy*zy)/zx*zx)	sin #2 color blending.
sin((zy*zy)/zx*zx)	sin #3 color blending.
atan(1/fabs(x*y))	atan #4 color blending.

11.7.1 Blend 00 command

linear scale command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.2 Blend 01 command

average (x + y)/2 command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.3 Blend 02 command

subtractive (x - y) command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.4 Blend 03 command

sum of squares 1 command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.5 Blend 04 command

sum of squares 2 command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.6 Blend 05 command

sin(x*y*100)*100 command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.7 Blend 06 command

atan(x/y)*50 command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.8 Blend 07 command

(x + y)/2 command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.9 Blend 08 command

log(fabs(1000/ Data->ytot))*700 command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.10 Blend 09 command

atan((x*x)/(y*y)) command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.11 Blend 10 command

atan(zx*zx/zy*zy) command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.12 Blend 11 command

sin(zx*zx/zy*zy) command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.13 Blend 12 command

sin(zx*zx/zy*zy) command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.7.14 Blend 13 command

atan(1/fabs(x/y)) command (Blend menu)

Color blending for regular or I see da light coloring mode. The selected formula is applied while mapping colors to pixels.

11.8 Color RGB command

RGB command (Color menu)

Use this command to use red/green/blue mapping, if in one of the 2D coloring modes(regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.9 Color RBG command

RBG command (Color menu)

Use this command to use red/blue/green mapping, if in one of the 2D coloring modes (regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.10 Color GRB command

GRB command (Color menu)

Use this command to use green/red/blue mapping, if in one of the 2D coloring modes (regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.11 Color GBR command

GBR command (Color menu)

Use this command to use green/blue/red mapping, if in one of the 2D coloring modes (regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.12 Color BRG command

BRG command (Color menu)

Use this command to use blue/red/green mapping, if in one of the 2D coloring modes(regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.13 Color BGR command

BGR command (Color menu)

Use this command to use blue/green/red mapping, if in one of the 2D coloring modes (regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.14 Color Triangle algorithm command

Triangle Algorithm command (Color menu)

When color values exceed the range of rgb components or palette indexes, the values are scaled with a triangle algorithm, or linear ramp.

11.15 Color Sine algorithm command

Sine Algorithm command (Color menu)

When color values exceed the range of rgb components, the values are scaled with Steven C. Ferguson's sine algorithm (non-palette modes only.)

11.16 Color Gray Scale command

Gray Scale command (Color menu)

Use this command to color the active image with gray tones, if in one of the 2D coloring modes (regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.17 Color Invert command

Invert command (Color menu)

Use this command to invert image colors, if in one of the 2D coloring modes (regular or I see da light.) If the image has been recently loaded, the image must be redrawn for this command to be enabled.

11.18 Color Divpal1 command

Divide by One Palette (Color menu)

Palette is not split before applying to pixel.

11.19 Color Divpal2 command

Divide by Two Palette (Color menu)

Palette is split into two parts before applying to pixel.

11.20 Color Divpal4 command

Divide by Four Palette (Color menu)

Palette is split into four parts before applying to pixel.

11.21 Color Divpal8 command

Divide by Eight Palette (Color menu)

Palette is split into eight parts before applying to pixel.

11.22 Color Palette mode command

Palette color mode command (Color menu)

Use this command to switch to palette-coloring mode. All plotting is mapped to a linear palette, which can be edited through the palette editor window.

11.23 Edit Palette

Edit Palettes

Calls up the Firefly palette editor, so you can modify the palette in use.

It is important to realize that palettes are software-simulated in Firefly (since 24-bit color supports no hardware palettes), so color-cycling and palette switching are not fast operations as with a 256-color system that supports palettes.

There are copy and spread options to smooth or customize the existing palettes in Firefly. You can then save all the palettes in a .pl file, or by saving the entire function and bitmap (v1.08+ saves all the palettes in the data file.)

Colors are shown in 8 groups of 29 colors, with four colors on the last row. This makes it easy to create divide-by-8, divide-by-4 and divide-by-2 palettes with 232 colors (Use a start color of 0 and an end color of 231.) With Firefly, a palette is actually 60160 colors, with each succeeding color (except the last) followed by 255 colors that are evenly spread from one color to the next.

Use the RGB-slider controls to edit any color in the palette. Select Copy to copy any color to another spot in the palette. Select Spread to define a smooth spread of colors from the current spot to another spot in the palette. Copy and Spread take effect immediately when you select another spot with the mouse button. You can cancel the operation with the Cancel button. In Firefly, colors do not cycle smoothly when you adjust the RGB/HSV sliders. This would be too slow with true color. The Map button is used to map color changes to an image after you are done adjusting the sliders. In the HSV mode, color spreads are based on HSV values instead of RGB values, which in some cases results in brighter color spreads.

Right-click on any point on the main window and the palette color for that pixel will be displayed in the palette editor. You can use any of the color-cycling keys (after clicking on the main window) to see the effects of the cycling in the palette editor window. Note: color cycling and color-selection-from-pixel only works when the image has been drawn in the current session. If you load a pre-existing image file, you must redraw it to cycle colors, etc. Anti-aliasing, undoing an action and hsv filtering also disable color cycling.

Use Reset to reset the colors of the palette in use, to where it was before it was cycled or modified. Note: if you change palettes with one of the function keys, any modifications to a previous palette are unaffected by the Reset button.

Use Reverse to reverse the order of the colors in the palette. This affects only those colors in the start-color to end-color range. Useful for reversing divide-by-eight palettes, etc., for orbit-trap pictures that require a reversed palette.

Use Neg to create a palette that is the complement of the current palette.

Use SRG to switch the red and green components of all palette colors.
Use SRB to switch the red and blue components of all palette colors. SRB and SRG are disabled in HSV mode. You can use these buttons to form eight different palettes by repeatedly switching red, green and blue components.

Use the Random palette button to randomize the current palette. The Randomize variables, rmin, rmax, bmin, bmax, gmin, and gmax act as limits that are applied after the palette after initial randomizing, to make the palette conform to the desired spectrum of colors.

11.24 Color-Scaling menu

Color-Scaling menu commands

The Color-Scaling menu offers the following commands:

Escape->	Escape-time color scaling.
Level	Color scaling based on log(z).
Continuous Potential	Color scaling based on continuous potential.
Use Level	Use level curve option for all coloring.
Background	Set external points to background color.

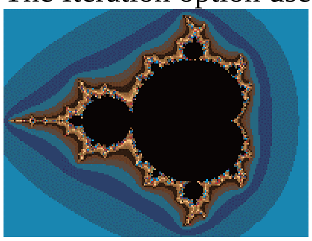
Set Only	Plot points in complex set only.
Graded Palette->	Use non-repeating vs modulus palette.
Use Palette 1 for Background Filter	Use palette 1 for background filter.

11.24.1 Escape

Escape

Five options are included that color a point based on its escape time (when it blows up.)

The Iteration option uses only the point's escape time.



Escape-time coloring.

The Iteration+ option uses the sum of a point's escape time and the value chosen (which can be picked from a menu that mirrors the Map menu.) A window is opened to set a q factor (1-200), which scales the sum value.

The Iteration* option uses the product of a point's escape time and the value chosen. .) A window is opened to set a q factor (1-200), which scales the product value.

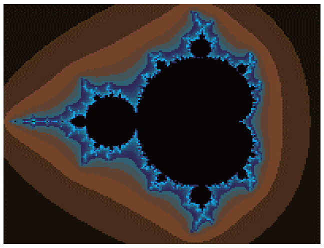
The Angle option use the absolute value of a point's exit angle (theta.) This is the atan method in Fractint.

The Angle-Iteration option uses the angle formed by the difference between a point's last two exit values and subtracts the point's escape time. Using the Angle-Only option on the Decomposition menu, escape times are not subtracted from the difference angle. This is Paul Carlson's atan method.

11.24.2 Level

Level

A point is colored based on its logarithmic escape. A window is opened to set a q factor, which controls the smoothness of picture color. A higher q factor results in grainier pictures and excess detail. Too low a q factor results in loss of colors and detail.

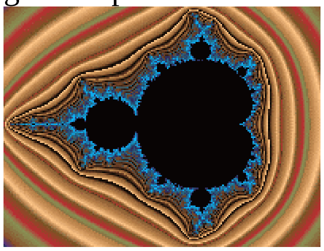


Level coloring.

11.24.3 Continuous Potential

Continuous Potential

A point is colored based on its continuous potential (when it blows up.) A window is opened to set a q factor, which controls the smoothness of picture color. A higher q factor results in grainier pictures and excess detail. Too low a q factor results in loss of colors and detail.



Continuous-Potential coloring.

11.24.4 Use Level Curve

Use Level Curve

All points are colored according to the choice selected from the Level-Curve Flag option. Defaults to Linear Map #4 if no Level Curve is checked. This works with Decomposition and other methods that would normally not use Level-Curve shading.

11.24.5 Background

Background

An external point is colored with the background color. This works like the Set Only flag, except with decomposition plots and Biomorph/Epsilon plots. Normally, when a point is decomposed, its escape time or level color is added to its arg (exit angle) to determine its final coloring. With Background color scaling, only a point's arg determines its color. With Biomorph/Epsilon plots, all external points are colored with the background color and all Biomorph/Epsilon points are colored with the set color.

11.24.6 Set Only

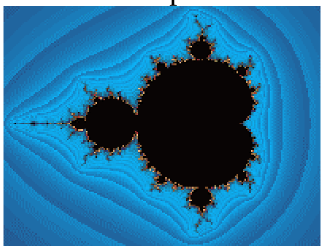
Set Only

The Set Only flag plots all external points in the background color.

11.24.7 Graded Palette

Graded Palette

All points are colored with a non-repeating graded palette versus the default repeating (modulus) color scaling. Has no effect on the Background or Use Level Curve options, or when you use the cutoff value in the Parameters window as a color multiplier. When used with the Escape/Iteration coloring mode and a negative cutoff value, iterations are interpolated to reduce banding in escape-time pictures. Three options are provided for smoothing. The Interpolated version works for most escape-time formulas except convergent types (Newton and renormalization.) The Mandelbrot version is based on Linas Vepstas' log log algorithm, and is designed mainly for formulas that use z^2 as their main focus, such as z^2+c. The Exponential smoothing method is based on Ron Barnett's algorithm, and works for both escape time and convergent-type fractals.



Level-graded coloring.

11.24.8 Use Palette

Use Palette 1 for Background Filter

This allows you to use palette 1 with a background filter, instead of the default palette, to add highlights to a picture. When Background filter is checked in the Filter window and

11.25 Palette menu

Palette menu commands

The Palette menu offers the following commands:

Palette #1-21Use one of 21 palettes.

11.26 Palette 1-21 command

Palette command (Palette menu)

Switch to palette #. Used with palette-coloring mode.

12 Convolve menu

Convolve menu commands

The Convolve menu offers the following commands:

Custom	Define a custom convolution filter.
Apply	Apply convolution filter.
Undo	Undo last filter action.
Average	Averaging filter.
Sharpen	Sharpening filter.
Edge Detection	Edge-detection filter.
Emboss	Embossing filter.
Blur	Blurring filter.

12.1 Convolve Dialog command

Convolution Dialog

Use this dialog to create and apply a custom filter to the active image. The convolution filter consists of a 3X3 matrix that can be varied to produce a sharpening, smoothing, or embossing filter, etc., when applied to the pixels of an image. The matrix represents the weight given to the target pixel (the center of the matrix) and its adjacent neighbors.

Click on Apply to apply the filter to the active image.

Click on Undo to reverse the effect of the filter.

For more information, see Graphics Gems II (edited by James Arvo) -- Image Smoothing and Sharpening by Discrete Convolution -- Dale A. Schumacher.

12.2 Convolve Apply command

Apply command (Convolve menu)

Use this command to apply the current filter in the convolution dialog to the active image.

12.3 Convolve Undo command

Undo command (Convolve menu)

Use this command to undo the effects of any convolution filter on the active image.

12.4 Convolve Average command

Average command (Convolve menu)

Use this command to apply an averaging filter to the active image.

12.5 Convolve Sharpen command

Sharpen command (Convolve menu)

Use this command to apply a sharpening filter to the active image.

12.6 Convolve Edge-detection command

Edge Detection command (Convolve menu)

Use this command to apply an edge-detecting filter to the active image.

12.7 Convolve Emboss command

Emboss command (Convolve menu)

Use this command to apply an embossing filter to the active image.

12.8 Convolve Blur command

Blur command (Convolve menu)

Use this command to apply a blurring filter to the active image.

13 View menu

View menu commands

The View menu offers the following commands:

Toolbar	Shows or hides the toolbar.
Status Bar	Shows or hides the status bar.

13.1 View Toolbar command

Toolbar command (View menu)

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in Firefly, such as File Open. A check mark appears next to the menu item when the Toolbar is displayed.

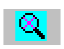
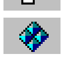
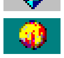
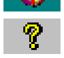
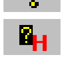
See Toolbar for help on using the toolbar.

13.1.1 toolbar

Toolbar

The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in Firefly,

To hide or display the Toolbar, choose Toolbar from the View menu (ALT, V, T).

Click	To
	Open the Remote which contains shortcut buttons for many common tasks and options in Firefly
	Open an existing drawing. Firefly displays the Open dialog box, in which you can locate and open the desired file.
	Save the active drawing or template with a new name. Firefly displays the Save As dialog box.
	Draw Mandelbrot set
	Draw Julia set
	Zoom into rectangle.
	Set image size.
	Adjust color controls.
	Edit palette.
	Edit formula/type data.
	Edit fractal parameters.
	Edit dll action list.
	Draw image from current parameters.
	Continue drawing.
	Reset coordinates.
	Show picture full-screen.
	Display info about Firefly.
	Display Firefly's help index.

13.2 View Status Bar Command

Status Bar command (View menu)

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

See Status Bar for help on using the status bar.

13.2.1 status bar

Status Bar

The status bar is displayed at the bottom of the Firefly window. To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar indicate which of the following keys are latched down:

Indicator	Description
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.

14 Window menu

Window menu commands

The Window menu offers the following commands, which enable you to arrange multiple images in the application window:

Cascade	Arranges windows in an overlapped fashion.
Tile	Arranges windows in non-overlapped tiles.
Arrange Icons	Arranges icons of closed windows.
Size Desktop	Size drawing area to window frame.
Window 1, 2, ...	Goes to specified window.

14.1 Cascade

Cascade command (Window menu)

Use this command to arrange multiple opened windows in an overlapped fashion.

14.2 Tile

Tile command (Window menu)

Use this command to arrange multiple opened windows in a non-overlapped fashion.

14.3 Arrange

Window Arrange Icons Command

Use this command to arrange the icons for minimized windows at the bottom of the main window. If there is an open drawing window at the bottom of the main window, then some or all of the icons may not be visible because they will be underneath this drawing window.

14.4 Size Desktop

Window Size DeskTop Command

Use this command to size the active drawing window to its frame size. Use after Tile command to reduce white space around a drawing that is smaller than screen size.

14.5 1,2, ...

1, 2, ... command (Window menu)

Firefly displays a list of currently open drawing windows at the bottom of the Window menu. A check mark appears in front of the drawing name of the active window. Choose a drawing from this list to make its window active.

15 A/V menu

A/V menu commands

The A/V menu offers the following commands:

Open AVI Stream	Open AVI file for writing and draw initial frame.
Write Frames	Write frames to AVI file.
AVI Variables	Edit AVI variables.
Close AVI Stream	Close an existing AVI stream.
View AVI	View an AVI animation file.
Start Midi	Enable midi sampling and set sample size.
End Midi	Disable midi sampling.
Save Midi	Save midi data to mdf and mid files.
Load Midi	Load midi variables from mdf file.

15.1 Open Avi Stream

Open Avi Stream…

Through a series of windows, this allows you to name and open an avi animation stream and choose a compression method. After choosing the frame rate (1-60) and using the file requester to name the file, you are given a choice of compression methods. You can also choose no compression for optimum view quality. (All compression methods degrade the original images, some more than others.) The first key frame in the stream is then drawn and written to the file.

Note: after the stream is opened, the size of the fractal that can be drawn is fixed at the size of the frame. No changes can be made to the size until the stream is closed.

15.2 Write Frames

Write Frames…

With this option, frames are written to a stream based on the difference between the current key frame and the previous key frame. The first key frame is written when you open a stream. The next key frame is created each time you use this option. In between you can zoom or change Avi variables as much as necessary. The stream is only written to when this option is used. The last key frame is automatically saved after the 'tween' series is written. The number of frames may range from 1-1500 frames between keys. With a frame number of 1 only the key frames are written. This allows animation to be created that incorporate all scalable variables in Firefly.

Use the Cancel button to exit this dialog without initializing a new series of frames.

Check the Log Scaling box if you want the frames to be written with logarithmic space between frames, else linear space is used. Useful when zooming, where frames would otherwise be packed together at the end of the frame series.

15.3 Avi Variables

Avi Variables…

This window contains all the major variables that Firefly now scales between key frames of an avi stream. They are identical to some of the variables found in the Parameters, Formula and Epsilon windows, plus a few other windows. If you decide to change a variable not included in this window, no scaling occurs. A few exceptions are the newlimit, limit, invert, torus, quaternion variables and the palette indexes (if you change palettes between key frames.) These are scaled also. If you change the basic function type, formula or color-scaling method, morphing isn't supported for these sorts of changes.

Note 1: when a formula is changed between key frames, the formula in the last key frame is used for tweening purposes. This may or may not produce useable images.

Note 2: when a frame number of 1 is used in the Write Frames window, all variables in FF that can be scaled are useable for animation. In this case the animation is composed of single key frames and optional tween frames.

15.4 Close Avi Stream

Close Avi Stream

Closes any open avi stream file. You need to do this before viewing the file or creating a new avi file. The stream is also closed when you exit Firefly.

15.5 View Avi

View Avi…

Opens an avi file for viewing. You can preview any multimedia file by clicking on its file name. A multimedia box will appear to the right of the file list. Click on okay to open the main view window.

There are buttons to Play a file forwards or Backwards, or forward automatically with Auto rewind/repeat. Click on Slow to slow down a video. Each click on Slow halves the viewing speed. A click on Stop freezes viewing and restores the view speed to normal playback.

Use the Open button to view a different avi file. Use the Save button to save the file in a different compression format. You must use a different name to save the file than the name that was used to open it. Click on the left-mouse button or any key to abort a save operation.

Note: the view avi requester can be used to preview any multimedia file, including midi files.

15.6 Start Midi..

Start Midi…

Start collecting sample data for the current fractal. This allows you to translate fractal data into a form that can be used later to create fractal music. You can specify the sample width for sampling (40-100.) This controls how many samples will be gathered and thus how large the midi file will be. The size of the current fractal must be larger than the sample size. A width of 40 creates a midi file length of about 3 minutes run time (at 24 clicks per division.)

15.7 Stop Midi..

Stop Midi…

Stop collecting sample data for the current fractal. You select this command to remove midi sampling from the iteration loop. Warning: you abandon any sample data collected when using this command.

Note: it's necessary to use this command sometimes when you need to use solid-guessing or anti-aliasing which are not available after starting midi sampling.

15.8 Save Midi

Save Midi…

Saves the sample data to a midi text file(c:\miditext.txt) then converts the text file to midi binary format. The format for the text file follows Piet van Oostrum's specifications for his text-to-midi converter, T2MF, which is used for this conversion. A mdf file is also created that saves the current midi parameters in this window and the sample size. The sample data is not saved. You need to recreate the sample data each time you reopen Firefly. But once the data is created, it can be modified again and again during the same working session.

After specifying a filename through the filename button and window, you can change any other parameter in the window to customize midi output. The divisions slider controls click per quarter note (1-100, default: 24), or the overall tempo of the music. You select 1-16 channels, and then apply a patch (instrument voice) to each channel. The pitch and volume can be adjusted separately for each channel (0-20, default: 5.) Since instruments tend to be pitch sensitive, min and max pitch controls are provided to tailor frequency response for each channel. Initial note (pitch), volume and clicks (time between notes) are set by choosing a filter that translates sample data for each parameter. The filter can be based on average color for a sample, last level or smallest level of a sample, the average exit angle or escape time for a sample, or a constant (default: 6*volume setting for each channel.)

15.9 Load Midi..

Load Midi…

Load midi parameters from a miff (midi-data) file. This allows you to recreate/edit a midi file using larger samples/different voices, etc.

16 Demo menu

Demo menu commands

The Demo menu offers the following commands, which illustrate various features of Firefly :

Random Julia(List)	Generate Julia fractal using random action list (plug-in based).
Random Julia2(Built-)	Generate random Julia fractal with built-in formulas.
Random Render (Normal)	Select a random rendering (normal coloring mode.)
Random Render (Palette)	Select a random rendering (palette coloring mode.)
Random Coloring (Normal)	Select a random coloring (normal coloring mode.)
Random Coloring (Palette)	Select a random coloring (palette coloring mode.)
Batch Mode	Repeat random fractal and save to file.

16.1 Random Julia (List)

Random Julia -- List (Demo menu)

A Julia fractal is generated from a randomly selected action list. Each list consists of a formula, rendering and filter. Post-rendering variables are also selected at random.

Related option: See the hot keys section for a description of the 'F' command.

16.2 Random Julia (Built-in)

Random Julia -- Built-in (Demo menu)

A random Julia fractal is generated using built-in formulas and rendering methods. Many of the built-in options of Firefly are selected on a random basis, and the Mandelbrot space for one of the hundred built-in formulas is scanned for an interesting Julia set. The palette used is also randomized. Note: In most case the Julia search is a short one, but sometimes the "seek" mode can seem to get stuck when the criteria for an interesting Julia set fails to match the formula used. In the latter case, either click the left mouse button and restart the search process.

Tip: some things remain to be done after the Julia set is drawn. The algorithm doesn't check the type of split palette that is used, so a mismatch in the "Divide by" color option may need correcting, e.g. Divide-by-four may be selected for a divide-by-eight palette. Feel free to experiment with all the parameters, reframe the image, change palettes etc. This routine provides a fast intro to many options in Firefly that the user may be unfamiliar with: no knowledge of fractal science/math required!

Related option: See the hot keys section for a description of the 'F' command.

16.3 Random Render (Normal)

Random Render -- normal (Demo menu)

The rendering options for the current fractal are randomized using variables specific to the normal coloring mode. Does not affect formula or range variables.

16.4 Random Render (Palette)

Random Render -- palette (Demo menu)

The rendering options for the current fractal are randomized using variables specific to the palette-coloring mode. Does not affect formula or range variables.

16.5 Random Coloring (Normal)

Random Coloring -- normal (Demo menu)

The coloring options for the current fractal are randomized using variables specific to the normal or generalized coloring modes. Does not affect formula or range variables.

16.6 Random Coloring (Palette)

Random Coloring -- palette (Demo menu)

The palette is randomized and coloring mode set to palette-based.

16.7 Batch Mode

Batch mode (Demo menu)

Here you set parameters for batching and saving random-generated images to disk. When the Repetitions value is non-zero, up to 1000 random images can be generated and saved to disk. Use a unique Filename to prevent batch files from overwriting existing image files. The Scan Limit directs the program on how many scans it makes through each formula before it skips to a new formula (if an interesting Julia fractal hasn't been found.)

A number of user-defined variables in the form of sliders and radio boxes have been added to customize random Julia searches. The Threshold sliders determine the iterative depth used to select a likely Julia set. Threshold1 provides the initial depth that a pixel must meet to continue in a particular area of Mandelbrot space. Then adjacent pixels are examined on the basis of limits set by Threshold2 and Threshhold3. Threshhold3 should be less than Threshold2. The Iterations slider sets the maximum depth of the pixel scan, analogous to the Iterations edit box in the Edit/Parameters window. You can choose to apply a random render to the fractal space, before or after searching, or not at all. The default button resets the Threshold, Iteration and Render variables to start-up values (No Render, Iterations=150, Threshold1=.5, Threshold2=.97, Threshold3=.5.) Note: A lower value for Threshhold1 tends to favor generation of fractals based on Newton's method, while a high value tends to favor producing more traditional escape-type fractals.

17 Help menu

Help menu commands

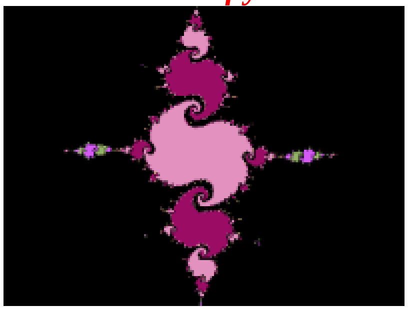
The Help menu offers the following commands, which provide you assistance with this application:

Getting Started	Tutorial for new users of Firefly.
Index	Offers you an index to topics on which you can get help.
Hot Keys	Quick reference to Firefly's hot keys.
Parser Info	Quick reference to Firefly's parser variables and functions.

17.1 Getting Started

Getting Started

Welcome to ***Firefly***!



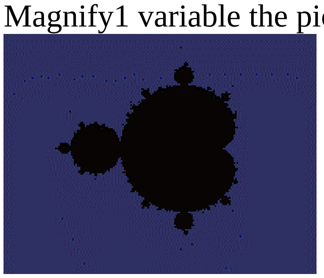
This is a short tutorial that will cover basic commands and background material necessary for a new user to create an initial picture with Firefly. For help on any menu command, press F1 while the command is highlighted. For help on the Edit DLL Formula/Render or Parameters window, click on the Help button inside that window.

The two methods of fractal composition that most fractal programs use are iterative and orbital. With the iterative method, which is used by Firefly, a formula is successively iterated until some criteria is met then the point is plotted. The process of iteration goes like this: starting from some initial value, a formula is evaluated, and then the result is used to reevaluate the formula. The formula is usually complex (based on complex variables of the form x+yi, where i=sqrt(-1)), the most commonly used formula being z^2+c. For two-dimensional fractals, the screen is mapped into ranges of the variables z or c. When the c variable is changed during iteration, the map produced is called the Mandelbrot set, after Benoit Mandelbrot. The z variable may be initialized at zero or some other value. When the z variable is changed during iteration, the c variable remaining fixed, a Julia set is formed. The Mandelbrot set is frequently considered a map of all Julia sets. If you choose a point for c anywhere on the border of the Mandelbrot set, a Julia set can be generated that looks very similar to the area it was taken from. The formula that came to be known as the Mandelbrot set, z^2+c, is only one of many complex formulas that can be used.

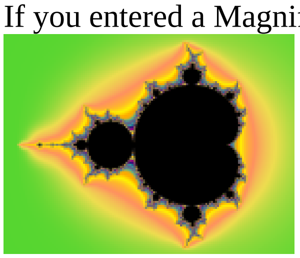
Since most fractaliers are initially interested in the Mandelbrot set, we will begin there. Firefly includes a basic Mandelbrot set in its default formula plug-in. If you open the Edit DLL Formula/Render window, the first formula in the formula listbox is the Mandelbrot set. First, clear out any current action list by clicking on the Delete button until no item shows in the Action listbox (Delete button will then be disabled.) Click on Add Formula to add the Mandelbrot formula to the action list. Firefly works on a list of actions for each image. They can be any grouping of formulas, renderings and filters. The order can be any order, but it usually works best to begin with a formula. The simplest image should also have a rendering method, so click on Add Render to add the first rendering method to the action list. When you click on any add button, the choice goes into the list immediately following the selection shown in the action list. Since the first rendering method is Escape Modulus, this requires an additional parameter to work well. Scroll to the Escape Modulus action, and enter 50 for the Magnify1 factor in the Render variables section. Escape Modulus uses Ron Barnett's exponential smoothing, so requires a color magnifier to enhance the color usage in this. Larger amounts of Magnify1 increase the palette usage. Using too high a magnification factor causes the palette usage to overlap top to bottom, with "banding" usually occurring. Click on Okay to exit the DLL edit window. A plot is automatically started, but click inside the active draw window to stop it, since the picture is not what we want yet.

Now reset the current image data and menus to their default values by selecting Reset/Figure in the Image menu. This changes the menus to Mandelbrot-type compatible settings. If you open the Parameters window through the Edit menu, you will notice a button near the bottom of the window labeled Okay. If you click on Okay (but not just yet), a two-dimensional plot of the Mandelbrot set will be drawn. First click on the Size slider carat and (holding the left mouse button down) drag it to the left until the value is something less than full-screen, say 250. Alternately, you can click to the left of the carat and the size will decrease by 4 with every click. Clicking on an end arrow changes the size by one. This works for all the sliders too.

Now click on Okay. Firefly's main window will be erased and a small plot of the Mandelbrot set will be drawn in the center of the Firefly window. If you didn't do anything with the Magnify1 variable the picture might look like something like below.



If you entered a Magnify1 value the picture should look something like below.



Go back to the Parameters window now and move the Sector to a value of 2. Press Okay. The right upper quarter of the window will be erased and the same Mandelbrot set will be drawn there. You can experiment with sector values 1-4 also. Each sector value puts the plot in a different quarter of the window. This works as long as the Size value is one-half of the horizontal size or less. A size above that forces Firefly to plot a picture in the center of the screen and change the sector to 0. You want to use sectors and smaller plot sizes to save time and when zooming out or framing.

Change the size to anything above the full-size (the image size set in the Size window) resolution, and then click on Okay. Firefly limits a plot size to full-size resolution so a rendition of the Mandelbrot set that fits the bitmap size will be drawn.

Clicking the mouse button inside the plot zone or pressing a key (other than the hot keys) stops the plot. You can continue the plot later by using the Continue command off the Image menu, or the Continue button in the Parameters window. Some commands disable the continue feature, as when changing to Solid-Guessing or Tesseral as a plot mode. These use a different plotting mode, incompatible with the default single-pass mode in Firefly. Note: instead of using plug-in formula/rendering options, the Mandelbrot set could also be drawn using the built-in formula P0(set in the Formula/Type window), and a built-in coloring option. You can also mix built-in formulas with plug-in rendering methods or vice versa... (The idea of this tutorial is to show the new plotting extensions available in Firefly.)
You'll probably be doing a lot of zooming and framing on your plots later, so we'll cover that briefly here. After the full-size version of the Mandelbrot set is finished (or as much of it is finished that you want to zoom in on), click the left-mouse button inside the draw window. When you move the mouse, a box a quarter the size of the window will appear that you can move around with the mouse. Hold the left-mouse button down to shrink the box, or the right-mouse button down to expand the box. Move the box over the area you are zooming in on, size the box if necessary and when it includes the details you want, press the space bar. The plot will be redrawn at zoom scale. To zoom out, you need to draw a smaller size plot, then zoom using a box larger than the plot drawn.

The second part of this tutorial involves creating Julia sets based on points inside a Mandelbrot set. There are two ways to generate Julia sets using Firefly. The manual way involves entering the complex constant values for a known Julia set into the complex c boxes in the Parameters window. The automatic way uses the "P" command, or keyboard hotkey. Draw a Mandelbrot set using MandelbrotP or Mandelbrot0 on the Type menu. Zoom into it until you find an area that would make an interesting Julia set. Press shift-P (Caps Lock off), and the cursor will change to a crosshatch. Position the cross hatch over the area of the Mandelbrot set you want to use as a starting basis for the Julia set. Click the left mouse button, and the pixel's coordinates will be entered into the complex c boxes as above. Use the Reset command on the Image menu to reset the Z ranges to full-scale. Change the type to Julia and redraw the picture. You should see a Julia set that closely resembles the area you picked from the Mandelbrot set. You can use the hotkey "J" to find Julia sets also. With this command, a small copy of the Julia set is immediately drawn in the second sector of the window each time you click on an area of the Mandelbrot set. When you find an interesting Julia set, click on the title bar or outside the window to exit this mode. Then change the Type to Julia and redraw the Julia set full-screen. Note: if you use the J command on an uninitialized image (one that was not redrawn after loading), the command will work but the original image of the Mandelbrot set will be erased, since no color info is available for it.

This completes the Getting Started tutorial. Be sure to read the hot keys section for additional info. The Bibliography lists additional reference material for a better understanding of the fractal types contained in Firefly.

17.2 Index

Index command (Help menu)

Use this command to display the opening screen of Help. From the opening screen, you can jump to step-by-step instructions for using Firefly and various types of reference information.

Once you open Help, you can click the Contents button whenever you want to return to the opening screen.

17.3 Hot Keys

Hot keys

Ctrl+F1-Ctrl+F9,Ctrl+F11, Ctrl+0-Ctrl+9 --- change to one of 21 color palettes -- useable during plotting.
Ctrl+F12 holds the palette of the most recently loaded function.

Regular or I see da light coloring mode:
Shift-F1 -- Color/Color Separation
Shift-F2 -- Color/RGB
Shift-F3 -- Color/RBG
Shift-F4 -- Color/GRB
Shift-F5 -- Color/GBR
Shift-F6 -- Color/BRG
Shift-F7 -- Color/BGR
Shift-F8 -- Color/Sawtooth Algorithm
Shift-F9 -- Color/Sine Algorithm
Shift-F11 -- Color/Gray Scale
Shift-F12 -- Color/Invert
Shift-1 -- Blend/00
Shift-2 -- Blend/01
Shift-3 -- Blend/02
Shift-4 -- Blend/03
Shift-5 -- Blend/04
Shift-6 -- Blend/05
Shift-7 -- Blend/06
Shift-8 -- Blend/07
Shift-9 -- Blend/08
Shift-0 -- Blend/09

Tab --- Replaces the currently selected palette with the palette in F11.
Useful when you want to make a palette file (.pl) from the palettes in a lot of individual bitmap files. Use the copy data and paste data commands to move the palette from another drawing window into F11. Select the palette(Ctrl+F1-Ctrl+F9,Ctrl+F12,Ctrl+0-Ctrl+9) you want to move Ctrl+F11 into, then press Tab.

up arrow --- forward cycle colors one step, including set color -- useable during plotting.
down arrow --- back cycle colors one step, including set color -- useable during plotting.

Shift-D -- use the mouse to examine pixel depth in a drawing. By clicking with the left-mouse button on any area in the current fractal, the pixel's depth and bailout values are displayed in the status bar, along with the pixel's coordinates. Useful to locate Mandelbrot islands while zooming on an image. Click on the title bar or press Esc to exit this command.

Shift-F -- generate a Julia set from a formula's MandelbrotP space. Random points in a formula's current Mandelbrot space are scanned for an interesting Julia set (enabled on when a Mandelbrot type is selected from the Type menu).

Shift-P --- grab point from Mandelbrot set (real and imaginary parts) and put values in complex constant. Cursor changes to a crosshatch, which you position over the area of the Mandelbrot set of interest. Then click the left-mouse button to transfer the pixel's coordinates to the c constant. Click outside window or in window frame to exit routine without "grabbing" a point.

Shift-J -- like "P", except that a Julia set is drawn immediately in sector 2 at size 100 and with iterations of 100. This is a fast exploratory routine for finding interesting Julia sets that can also be used where the Mandelbrot set is discontinuous as in the Phoenix formula. Unlike the "P" command, "grabbing,"(and drawing) continues until you click on the window's frame. Note: when you exit the J command, once you find an interesting Julia set, another window is opened with the Julia type set. The parameters in the original window revert to their original Mandelbrot settings.

Shift-C -- clear the screen to the current background color.
Shift-B -- erases the 3D background, leaving the 3D plot intact.

Shift-Z -- zoom in/out coordinates. Like the menu command except does not immediately redraw the picture. This allows you to zoom into another screen sector without erasing the previous picture.

Shift-T -- annotate a picture with text. Cursor changes to a crosshatch, which you position over the area where you want the text to start. Then click the left-mouse button to transfer any text (from the Edit/Text window) to the picture. Can be used with Undo. Use the Edit/Text command to change font, text color or format text into multiple lines. This is useful for adding copyright/author info to a finished picture.

17.4 Parser

Parser Information

Functions (capital letters are optional, and parenthesis are necessary around complex expressions)

The following information takes the form "standard function" ---"form used by Firefly to represent standard function".

sine z --- sin(z) or SIN(Z) ; where Z can be any complex expression
hyperbolic sine z --- sinh(z) or SINH(Z)
arcsine z --- asin(z) or ASIN(Z)
cosine z --- cos(z) or COS(Z)
hyperbolic cosine z --- cosh(z) or COSH(Z)
arccosine z --- acos(z) or ACOS(Z)
tangent z --- tan(z) or TAN(Z)
hyperbolic tangent z --- tanh(z) or TANH(Z)
arctangent z --- atan(z_) or ATAN(Z)
cotangent z --- cotan(z) or COTAN(Z)
arccotangent z --- acotan(z) or ACOTAN(Z)
e^z --- exp(z) or EXP(z) -- the exponential function
natural log of z --- log(z) or LOG(Z)
absolute value of z --- abs(z) or ABS(Z)
square root of z --- sqrt(z) or SQRT(Z)
z squared --- sqr(z) or SQR(Z)
real part of z --- real(z) or REAL(Z)
imaginary part of z --- imag(z) or IMAG(Z)
modulus of z --- mod(z) or MOD(Z) or |z| -- (x*x + y*y)
conjugate of z -- conj(z) or CONJ(z) -- (x-yi)
flip(z) --- flip(z) or FLIP(Z) -- exchange real and imaginary parts of z (y+xi)
polar angle of z -- theta(z)

if/then/endif – if(argument), then (phrase) endif -- if argument is true then do phrase else skip phrase('then' tag is optional, but a comma should follow argument or put 'if(argument)' on separate line)

if/then/else/endif - if(argument), then (phrase) else (phrase) endif -- if argument is true then do phrase else skip phrase and do alternate phrase('then' tag is optional, but a comma should follow argument or put 'if(argument)' on separate line)

Note: if/then/endif and if/then/else/endif loops can be nested only when endifs follow each other at the end of the loops. For example: if(argument) if(argument) then (phrase) endif endif.

Math operators

+ --- addition
- --- subtraction
* --- multiplication
/ --- division
^ --- power function
< --- less than
<= --- less than or equal to

> --- greater than
>= --- greater than or equal to
!= --- not equal to
== --- equal to
|| --- logical or (if arg1 is TRUE(1) or arg2 is TRUE)
&& --- logical and (if arg1 is TRUE and arg2 is TRUE)

Constants and variables

complex constant --- c# or C#, read/write.
complex conjugate --- cc# or CC#, read-only.
convergence limit --- cl# or CL# -- the constant entered in the Converge gadget, read-only.
cr -- the constant entered in the cr box in the Parameters window(use j# for parser)
ci -- the constant entered in the ci box in the Parameters window(use k# for parser)
e --- e or E -- 1e^1 -- 2.71828, read/write.
i --- i or I -- square root of -1,read/write.
iteration --- iter# -- iteration loop counter
j --- j# or J# -- real part of the complex constant, read-only.
k --- k# or K# -- coefficient of the imaginary part of the complex constant, read-only.
Note: j and k are the actual values of the complex constant terms as they are used in the iteration process, so will vary when the Mandelbrot option is used.
m --- m# or M# or pixel --a complex variable mapped to the pixel location as defined by the z coordinates entered in the Parameters window, read/write.
maxit -- the maximum number of iterations, as set in the Parameters window, read only
p --- p# or P# -- real constant used in phoenix maps; uses the real part of the complex constant when the Phoenix option is chosen, read-only.
p1 – the complex constant entered in the cr and ci gadgets, read-only.
pi --- pi or PI -- 3.14159, read/write.
q --- q# or Q# -- real constant used in phoenix maps; uses the imaginary part of the complex constant when the Phoenix option is chosen, read-only
x --- x# or X# -- real part of Z, read/write.

y --- y# or Y# -- coefficient of the imaginary part of Z, read/write.

z --- z or Z -- function value at any stage of the iteration process, read/write.
zn# or ZN# -- the value of z at the previous stage of iteration, read-only.

17.5 Built-in Formulas

Built-in Formulas (enter the following prefix into the Function #1 or Function #2 edit boxes)

p0 -- z^2+c --- the standard Mandelbrot or Julia set.
p1 -- cz(1-z) --- the self-squared dragon set.
p2 -- c(z-1/z) --- alternate Mandelbrot or Julia set.
p3 -- cz^2-1 --- alternate Mandelbrot or Julia set.

p4 -- c^2/(c+z^2) --- alternate Mandelbrot or Julia set.
p5 -- z^3+c --- cubic Mandelbrot or Julia set.
p6 -- ((z^2+c-1)/(2z+c-2))^2 -- renormalization formula #1 for x-plane or q-plane pictures (Note 9).
p7 -- z^2+j+kzn --- Phoenix curve (Ushiki). Uses Fractint extensions for degree(>2 or <-3.).
p8 -- Julia/Mandelbrot set (modified from M. Barnsley) (Note 2).
p9 -- fn(z)-cfn(z) -- generalized frothy basin (J. Alexander.) (Note 7).

r0 -- Newton/Halley map of z^3+conj(z)c -- exploratory function based on modified frothy basin.
r1 -- z^z+z^s+c --- Biomorphs, etc.; s may be complex using the si variable as its imaginary component.

r2 -- z^s-z+c --- Biomorphs, etc.; s may be complex using the si variable as its imaginary component.

r3 -- fn(z)+exp(z)+c -- Biomorphs, etc.
r4 -- solves Newton/Halley transformation of (z^2-c)(z+1). (Notes 3,4,5,11).
r5 -- cfn(z) -- transcendental Julia curve, etc.
r6 -- cexp(z) -- exponential Julia curve, etc. with additional plane checking when real value of Z exceeds 50. If cos(imag-Z) >= 0, point is considered part of Julia set.
r7 -- fn(z)+cfn(z)+1 -- generalized form of t9.
r8 -- foggy coastline #1 Mandelbrot IFS (M. Barnsley) (Note 14).
r9 -- foggy coastline #2 Mandelbrot IFS (M. Barnsley) (Note 14).

e0 -- solves Newton/Halley transformation of (z+j)(z+k)(z^2+1) for either Julia or Mandelbrot set.
e1 -- solves Newton/Halley transformation of (z+j)(z^2+z+k) for Mandelbrot and Julia set.
e2 -- solves Newton/Halley transformation of (z-1)(z^2+z+c) for either Julia or Mandelbrot set.
e3 -- solves Newton/Halley transformation of (z+j)(z+k)(z+1) for either Julia or Mandelbrot set.
e4 -- Chaos Game Julia IFS (M. Barnsley).
e5 -- snowflake Julia IFS (as described in Fractals Everywhere by M. Barnsley).
e6 -- solves Newton/Halley transformation of logz-c.
e7 -- solves Newton/Halley transformation of exp(z)-c.
e8 -- solves Newton/Halley transformation of (z-c)(z+1)(z-1) for Mandelbrot or Julia set.
e9 -- solves Newton/Halley transformation of (z-c)(z+c)(z^2+c^2) --- z^4-c^4.

s0 -- solves Newton/Halley transformation of sinz-c.
s1 -- sexpz+c -- transcendental Mandelbrot or Julia set.
s2 -- c(1+z^2)^2/(z^2-1) -- alternate Mandelbrot/Julia set.
s3 -- solves Newton/Halley transform of tan(z)-c .
s4 -- IFS (x=sy+j,y=-sx+k (x>0); else x=sy-j,y=-sx-k (modified from M. Barnsley) (An alternate version of this formula is executed when the limit value is non-integral.)
s5 -- solves Newton/Halley transform of z^s-1 (Julia set only).
s6 -- composite function cz-c/z & z^2+c (C. Pickover).

s7 -- transcendental function fn(z)+c .
s8 -- ((z^3+3(c-1)z+(c-1)(c-2))/(3z^2+3(c-2)z+c^2-3c+3))^2 -- renormalization formula #2 for x-plane or q-plane pictures .
s9 -- Newton/Halley map of z(z^limit-1) (Julia set only).

t0 -- Newton/Halley map of z(z^limit-c) (Julia or Mandelbrot set ; display method 2 default; limit>=1.0 ; use 1.0 for initial z with Mandelbrot0 type, or use MandelbrotP type.).
t1 -- Newton/Halley map of Chebyshev function cos(n*arccos x).
t2 -- Newton/Halley map of Hermite polynomial: 16x^4-48x^2+12.
t3 -- alternate Newton/Halley map of tanz-c (Julia or Mandelbrot set.) the twist is in the second derivative of the Halley type.
t4 -- Newton/Halley map of z^limit-c (Julia or Mandelbrot set ; display method 2 default; limit>=1.0 ; use 1.0 for initial z with Mandelbrot0 type, or use MandelbrotP type.).
t5 -- fn(fn(z))+c -- user-defined complex set. When the first function is z^2 and the second function is conj(z), this becomes the z-conjugate set, zz^2+c, the tricorn set. (Note 12).
t6 -- Volterra-Lotka equations discretized by modified Huen method (from The Beauty of Fractals).
t7 -- c^z+c -- tetration of z.
t8 -- q^2+c -- Quaternion set (from Computer, Pattern, Chaos and Beauty) (Note 8).
t9 -- z+cz+1 --try with Newton's method applied. A buggy algorithm found this one.

a0 -- spiral network -- C. Pickover.
a1 -- z-(1/z+c) -- try with renormalization applied. Sequel to t9.
a2 -- fn(z)-(fn(z)+c) -- generalized form of a1.
a3 -- alternate Newton/Halley map of sinz-c. see t3 variant.
a4 -- user-defined complex set: fn(z)+fn(z)+c.
a5 -- hypercomplex Newton/Halley map of h^3+c.
a6 -- Hypercomplex Newton/Halley map of fn(h)+c.
a7 -- user-defined complex set: fn(z)+fn(c).
a8 -- fn(z)+zn+c -- from Fractal Creations.
a9 -- q^3+c -- cubic Quaternion set.

b0 -- alternate Newton/Halley map of exp(z)-c. as for t3 variant.
b1 -- alternate Newton/Halley map of log(z)-c.
b2 -- Newton/Halley map of phoenix curve.
b3 -- cfn(z)+zn -- user-defined complex formula.
b4 -- fn(z)+kzn+j -- generalized phoenix curve formula.
b5 -- fn(z) a preformula for use with type 3 composite fractals. uses limit gadget to select function.
b6 -- Newton/Halley map of fn(z)+fn(z)+c.
b7 -- Newton/Halley map of cfn(z).
b8 -- fn(z)*fn(z)+c.
b9 -- Newton/Halley map of foggy coastline #1.

c0 -- Newton/Halley map of foggy coastline #2.
c1 -- Newton/Halley map of fn(fn(z))+c.
c2 -- cfn'(z), where fn'(z)=first derivative of user-defined function.

c3 -- fn(z)+fn'(z)+c.
c4 -- fn'(z)+fn(c).
c5 -- fn(fn'(z))).
c6 -- first order gamma function: (z/e)^z*sqr(2*pi*z)+c.
c7 -- Newton/Halley map of fifth degree Legendre polynomial: 1/8(63z^5-70z^3+15z; display method 2 default.
c8 -- (z^2+e^(-z))/(z+1): second-order convergence formula for finding root of ze^z-1=0.
c9 -- Newton/Halley map of fn(z)*fn(z)+c.

d0 -- z^s/limit+c: anti-derivative of z^n; s may be complex using the si variable as its imaginary component.
d1 -- Sterling expansion of gamma function: (z/e)^z*sqr(2*pi/z)+c.
d2 -- Newton map of fn'(z)-fn(z)+c: generalized first degree Laguerre polynomial. Newton map only. (Note 13).
d3 -- fn(1/(fn(z)+c)).
d4 -- z^2-c; where zreal=abs(zreal)(Paul Carlson's "alien" Julia set).
d5 -- z^2; where zreal=abs(zreal)-cr, zimag=zimag-ci (Paul Carlson Julia set).
d6 -- cfn(z)+c.
d7 -- Newton's method applied to (x^3+y^2-cr=0 and y^3-x^2+ci=0). Newton map only. from Sylvie Gallet and Fract19.par.
d8 -- Newton's method applied to fn1(x)+fn2(y)-cr=0 and fn3-fn4+ci=0 (Note 15) Newton map only.
d9 -- Bill13 from Bill Rossi via the Internet.

f0 -- generalized form of Earl Hinrichs' sophomore sine function(ssin) -- limit*fn(z)+s+si, where fn(z)=(fn1(x),fn2(y)).
f1 -- c^2/(1-cz^2) -- variant of p4.
f2 -- Gallet-4-01, from Sylvie Gallet's extensive Internet collection (Note 16).
f3 -- Gallet-4-02, from Sylvie Gallet.
f4 -- Gallet-6-01, from Sylvie Gallet.
f5 -- Gallet-6-02, from Sylvie Gallet.
f6 -- Gallet-6-03, from Sylvie Gallet.
f7 -- Gallet-6-04, from Sylvie Gallet.
f8 -- Gallet-6-05, from Sylvie Gallet.
f9 -- Gallet-7-01, from Sylvie Gallet.

g0 -- z^3-3c^2z+s -- cubic Mandelbrot (Note 17)
g1 -- z^3-3sz+c -- alternate cubic Mandelbrot
g2 -- z^3-3c^2z^2+s -- cubic Mandelbrot variant
g3 -- z^3-3sz^2+c -- alternate cubic Mandelbrot variant
g4 -- z^3-3c^2/z+s -- cubic Mandelbrot variant
g5 -- z^3-3s/z+c -- alternate cubic Mandelbrot variant
g6 -- z^3-3c^3*z+s -- cubic Mandelbrot variant
g7 -- z^3-3s/z^2+c -- alternate cubic Mandelbrot variant
g8 -- z^3-3c^2+z+s -- cubic Mandelbrot variant
g9 -- z^3-3s+z+c -- alternate cubic Mandelbrot variant

h0 -- (O*C^-1)(C*O)+c -- octonion set (Note 18)
h1 -- cO*CC(1-C*O) -- octonion set
h2 -- O*C(O-CC*O)+c -- octonion set
h3 -- cO^2+O^2*CC-c -- octonion set
h4 -- O^2*CC+O^2*C+c -- octonion set
h5 -- O^3*CC+c -- octonion set
h6 -- O^3*CC+O^3*C+c -- octonion set
h7 -- O^4*CC+c -- octonion set
h8 -- cO^3*CC+c -- octonion set
h9 -- O^2*CC+O^3*C+c -- octonion set

i0 -- 2*z*c#*cos(pi/z) -- Godwin Vickers
i1 -- 2*z*c#*sin(pi/z) -- Godwin Vickers
i2 -- 2*z*c#*tan(pi/z) -- Godwin Vickers
i3 -- 1/z^3+sin(z)*c^2
i4 -- cosh(z)/c*z+c^2
i5 -- exp(z)/z^3-c
i6 -- cosh(z)*z^2-c^2
i7 -- 1/z^2-cz-c
i8 -- tan(z)-czc
i9 -- (tan(z)-1/z^3)/c^2

j0 -- cosh(z)*cos(z)+1/c
j1 -- z^4/(z^3-c^2)
j2 -- 1/z^2-tan(z)+c
j3 -- tan(z)/z^3+c
j4 -- 1/z^3-cz+1/c
j5 -- z^3+tan(z)*c
j6 -- 1/z^3-tan(z)-c
j7 -- tan(z)-sin(z)+c
j8 -- 1/z^2-tan(z)+1/c
j9 -- z^4+sin(z)/c

k0 -- Mandelbrot set(sine variation)
k1 -- z^1.5+c -- Godwin Vickers
k2 -- (z^2-z^(2-s))/s+c -- Escher set by Roger Bagula
k3 -- z^2+z/(|z|+c) -- Roger Bagula
k4 -- quantum set -- S.M. Ulam
k5 -- prey predator #1 -- Roger Bagula
k6 -- prey predator #2 -- Roger Bagula
k7 -- Klein group #1 -- Roger Bagula
k8 -- Klein group #2 -- Roger Bagula
k9 -- Klein group #3 -- Roger Bagula

L0 -- Loxodromic by Thomas Kromer (fixed type)
L1 -- squared loxodrome
L2 -- Gedatou by Thomas Kroner (fixed type)

L3 -- Ventri by Thomas Kroner (fixed type)
L4 -- squared gedatou
L5 -- fn(z)-cfn(z)
L6 -- fn(z)+fn(z)+c"
L7 -- cfn(z)+c"
L8 -- fn(z)+cfn(z)+1"
L9 -- fn(z)+c

m0 -- fn(fn(z))+c
m1 -- fn(z)+fn(c)
m2 -- fn(z)+zn+c
m3 -- cfn(z)+zn
m4 -- fn(z)+kzn+j -- generalized phoenix curve
m5 -- fn(z)*fn(z)+c
m6 -- fn(1/(fn(z)+c))
m7 -- (1/fn(z))^2+c
m8 -- (1/fn(z))^3+c
m9 -- fn(z)/(1-fn(z))+c

n0 -- Sinus by Thomas Kromer (fixed type)
n1 -- Sinus #2 by Thomas Kromer (fixed type) (Note 19)
n2 -- Rings of Fire by Thomas Kromer (fixed type) (Note 19)
n3 -- Teres by Thomas Kromer (fixed type)
n4 -- z^2+y+c
n5 -- z^2+y[n+1]+c
n6 -- z^2+zi+c
n7 -- z^2+zi[n+1]+c
n8 -- zr^2+3zi+c
n9 -- zr^2+4zi+c

Note 1: all pertinent menu flags must be set for built-in functions to work as described.

Note 2: For further info on Michael Barnsley's formulas, see his "Fractals Everywhere".

Note 3: Halley map requires the Newton flag to be set. This is another numerical approximation method for finding complex roots. For all Newton/Halley functions, the Newton map is the default. The Halley option is specified through the Arg Gadget, the second character being set to 'h', after the display method (1-9). E.g. '1hr' would designate a relaxed Halley map with display method 1.

Note 4: Halley and Newton maps can use one of nine display methods:
#1 (the default mode, except for functions using sinz, expz, logz and tanz, or fn(z), which default to method 2, and don't use methods 1,4 or 5): ---colors represent the root (the zero), which a point converges to.
#2 (if the Arg Gadget is set to 2, or for functions of sinz, tanz, logz, expz, and fn(z)): ---colors represent the number of iterations a point takes to converge.
#3 (if the Arg Gadget is set to 3) -- colors represent the number of iterations a point takes to

converge according to an alternate formula described by C. Pickover in Computers, Pattern, Chaos and Beauty.
#4 (if the Arg Gadget is set to 4) -- a merging of methods 1 and 3. After the point converges according to the alternate formula #3, its roots are colored according to #1.
#5 (if the Arg Gadget is set to 5) -- a variation of method 1, with double-convergence checking inside the loop.
#6, #7 and #8-- alternate convergent formulas.
#9 -- a variation of method 3, with double-convergence checking.

Note 5: An additional third argument that affects the convergence speed of Newton/Halley maps may be one of the following five methods:
'r': relaxed Newton method uses the formula z = z-sf(z)/f'(z).
'm': modified Newton transformation uses the formula: z-(f(z)/(f'(z)+si)). Note: Si here references the s variable * i, not the complex variable s+si.
'd': relaxed modified Newton method uses the formula: z-(sf(z)/f'(z)+si)).
'p': premodified Newton transform uses the formula: sz-(f(z)/f'(z)).
'c': complex Newton transform uses the formula : z-(f(z)/(f'(z)+c)), where c is the complex constant.
The s constant is entered via the S gadget.

Note 7: the term 'fn(w)' represents any one of 47 user-defined functions chosen through the f1-f4 gadgets:
0: sin(w). 1: sinh(w). 2: cos(w). 3: cosh(w).
4: tan(w). 5: tanh(w). 6: exp(w). 7: ln(w).
8: w^c 9: w^z. 10: 1/w. 11: w^2.
12: w^3. 13: abs(w). 14: sqrt(w). 15: w.
16: conj(w). 17: csc(w). 18: csch(w). 19: sec(w).
20: sech(w). 21: cot(w). 22: coth(w). 23: cw.
24: 1. 25: arsin(w). 26: arcsinh(w).
27: arccos(w). 28: arccosh(w). 29: arctan(w).
30: arctanh(w). 31: arccot(w). 32: arccoth(w).
33: vers(w). 34: covers(w). 35: L_3(w): 3rd degree Laguerre polynomial. 36: gamma(w): first order gamma function.
37: G(w): Gaussian probability function -- (1/sqr(2pi))*e^(.5w^2).
38: c^(s+si). 39: zero. 40: w^(s+si). 41: |(wx)|+|(wy)|*i(abs).
42: wy+wx*i(flip). 43: conj(cos(w))--cosxx. 44: theta(w) -- polar angle(w).
45: real(w). 46: imag(w).
When only fun#1 or fun#2 is used and a single user-defined function is involved, the function is taken from f1. When two user-defined functions appear in a function, the f2 gadget supplies the second function type, except as noted below. For Newton/Halley maps involving z^z, the first derivative is defined as z^z*(1+ln(z)). An alternate derivative formula (z*z^(z-1)) is used when a non-integral value is entered as an arg limit (e.g.: 0.1). (This produces interesting effects, though mathematically inaccurate.) For plots that use both fun#1 and fun#2(type 2 or 3, etc), fun#1 takes its functions from f1 and f2 and fun#2 takes its functions from f3 and f4.

Note 8: The quaternion and hypercomplex functions use the complex c gadgets to input cr, ci, cj and ck. These may be zero when generating a Mandelbrot-like set of these functions. Julia sets may then be mapped by grabbing points (cr,ci) from interesting areas near this set. Cj and ck must be entered manually for Julia sets. The hj and hk gadgets are used to input the z and w coefficients of the j and k planes. Use small amounts to start for these variables (0-1.0.) Values of 0 for hj, hk, cj and ck result in a two-dimensional slice that matches the standard (non-hypercomplex) type. Higher values of z and w (as well as cj and ck) produce more pronounced asymmetry in the complex mapping.

Note 9: Renormalization functions use the Arg Gadget for plotting options (1-4,6-8) as follows:

0 or 1: default renormal, with anti-ferromagnetic points mapped only for Julia sets. Paramagnetic points (those converging to 1) and ferromagnetic points (those escaping to infinity) are mapped for both Mandelbrot and Julia sets.

2: anti-ferromagnetic points are mapped for the Mandelbrot set. This is actually a level-set mapping for points that do not escape to infinity or converge to 1.

3: uses an alternate convergence formula for paramagnetic and anti-ferromagnetic points.

4: a combination of methods 0 and 3, with characteristics of both methods appearing in plot.

6-8: alternate convergence methods, same as those used with Newton/Halley maps

An optional argument for renormalization 'n' follows the convergence method. This is an alternate bailout method for ferromagnetic points.

Note 10: Most of the built-in functions (except for real Newtons and the Gallet formulas) have hypercomplex extensions when values of cj, ck, hj or hk are non-zero.

Note 11: Hypercomplex Newton/Halley maps use only type 2 and type 3 convergence tests.

Note 12: The default version of hypercomplex conjugate is defined as conjugate(h)=hr-hi-hj+hk. A variant of the hypercomplex conjugate uses an arg limit with a non-integral value (e.g.: 2.1.) This makes all imaginary components of h negative, such that conjugate(h)=hr-hi-hj-hk.

Note 13: The formula for a first degree Laguerre polynomial is e^t(d/dt(t/e^t))=d/dt(t)-t.

Note 14: With a non-integral value entered in the limit gadget, an alternate (Fractint) version of this formula is executed.
Note 15: For real Newtons, the function selected from the f1-f4 boxes is a real function. For a real conjugate, the negation of the real term is used.
Note 16: formulas by Sylvie Gallet have been modified to allow both Mandelbrot and Julia sets to be drawn from them. Except for Gallet-6-02, the bailout is set with the built-in variable bailout in the Parameters window. Gallet-6-02 uses the complex constant p3 (limit and converge) for bailout.
Note 17: For the traditional cubic Mandelbrot (example in The Science of Fractal Images), hj, hk, cj and ck (hypercomplex components of z and c) should be set to zero.

Note 18: Octonions have a form of xr+xi+xj+xk+xE+xI+xJ+xK. For these formulas C is the octonion constant (1,1,1,1,1,1,1,1) and CC is the octonion conjugate (1,-1,-1,-1,-1,-1,-1,-1). Additional options are entered via the Arg box in the Formula editor window. To rotate the extra four-octonion dimensions (E-K) use the following syntax:

0I	--	rotate OE-OK to OI-OE
0J	--	rotate OE-OK to OJ-OI
0K	--	rotate OE-OK to OK-OJ

To normalize the C and CC constants:

n	--	normalize C and CC (default is un-normalized)

Alternate octonion initialization:

c	--	set OE-OK to .01 at beginning of each iteration

Note 19: For the loxodromic functions, Sinus #2 and Rings of Fire, the Arg Limit variable (in the Edit Formula/Type window) is used as an additional ingredient. Try values -1.5 to 1.5 for Sinus #2 and 1.5 or PI/2 for Rings of Fire.

17.6 Help Default Plug-ins command

Default Plug-ins (Help menu)

Firefly uses three default plug-ins(dynamic libraries), "fractal.fll", "fractal.rll" and "fractal.fil".

Fractal.fll is the formula library. It currently contains 34 formulas. This is similar to the built-in formula set (P0-F9) that is available through the Edit/Formula/Type window. All formulas except #33 use hypercomplex (h) extensions when the flag is activated. This occurs whenever cj, ck, hj or hk is non-zero. In addition, most formulas have quaternion(q) extensions that are activated whenever a quaternion is drawn using these formulas and the "Use quaternion math" box is checked. A description of the default formulas and extensions available through their Configuration dialog follows:

0 -- z^2+c -- standard Mandelbrot set. h(hypercomplex),q(uaternion)
1 -- cz(1-z) -- the self-squared dragon set. h,q
2 -- c(z-zr) -- alternate Mandelbrot set. h,q
3 -- cz^2-1 -- alternate Mandelbrot set. h,q
4 -- c^2/(c+z^2) alternate Mandelbrot set. h,q
5 -- z^3+c -- cubic Mandelbrot set. h,q
6 -- ((z^2+c-1)/(2z+c-2))^2 -- normalization formula #1. h,q
7 -- z^2+j+kzn -- Phoenix formula(Ushiki) h,q
8 -- (x^2-y^2-1,2xy) when x>0; else (x^2-y^2-1+jx,2xy+ky) -- IFS by Michael Barnsley. h only
9 -- solves Newton/Halley transform of z^3+conj(z)c -- modified frothy basin. h,q,n(ewton) Note 1
10 -- z^z+z^se+c -- Biomorphs, etc. h,q,e(xponential) Note 2
11 -- z^se-z+c -- Biomorphs, etc. h,q,e
12 -- cexp(z) -- exponential Julia curve. h,q
13 -- (z-1)/c when x>=0; else (z+1)/cc -- foggy coastline #1 by Michael Barnsley. h,q
14 -- (z-1)/c when kx-jy>=0; else (z+1)/c -- foggy coastline #2 by Michael Barnsley. h,q
15 -- x=2x,y=2y-1 (y>.5); x=2x-1,y=2y (x>.5); else x=2x,y=2y -- Chaos Game Julia set by Michael Barnsley. h only

16 -- snowflake -- IFS by Michael Barnsley. h only
17 -- (solves Newton/Halley transform of) logz-c. h,q,n
18 -- (solves Newton/Halley transform of) exp(z)-c. h,q,n
19 -- (solves Newton/Halley transform of) sin(z)-c. h,q,n
20 -- se*exp(z)+c -- transcendental Mandelbrot or Julia set. h,q,e
21 -- (c(z^2+1)^2)/(z^2-1) -- alternate Mandelbrot/Julia set. h,q
22 -- (solves Newton/Halley transform of) tan(z)-c. h,q,n
23 -- strange attractor(when se=1.4142) -- IFS by Michael Barnsley. h,q,e
24 -- cz-c/z && z^2+c -- composite function by Clifford Pickover. h,q
25 -- ((z^3+3(c-1)z+(c-1)(c-2))/(3z^2+3(c-2)z+c^2-3x+3))^2 --Normalization formula #2. h,q
26 -- alternate (*Halley* map of) tanz-c. h,q,n
27 -- (Newton/Halley map of) z^c+c. h,q,n
28 -- discretizes Volterra-Lotka equations via modified Heun algorithm. h,q
29 -- c^z+c -- tetration formula. h,q
30 -- z+cz+1 -- by Terry W. Gintz. h only
31 -- spiral network -- by Clifford Pickover. h,q
32 -- z-(1/z+c) -- by Terry W. Gintz. h,q
33-- z^2+c -- assembly language optimization of standard Mandelbrot set by Damien Jones -- log log smoothing by Linas Vepstas. m(agnify) Note 3

Fractal.rll is the rendering library. It currently contains 21 rendering actions. Rendering is used to set iteration-loop bailout conditions, which affect coloring on a pixel by pixel basis. A description of the default renderings and extensions available through their Configuration dialog follows:

0 -- Escape Modulus -- Standard Escape method for Mandelbrot/Julia sets; break loop when mod(z)>bailout -- Exponential smoothing by Ron Barnett. m
1 -- Biomorph -- From Clifford Pickover; break loop when either zreal or zimag>bailout. m
2 -- Orbit trap -- epsilon cross -- Derived from Clifford Pickover's Epsilon Cross method, with psuedo 3d extensions by Paul Carlson. o(rbit), o2(orbit 2), z(pen),ex(clude) Notes 4,5,6,7
3 -- Orbit trap -- globe -- A variation of Paul Carlson's pseudo 3d method, by Terry W. Gintz. o,o2,z, ex
4 -- Orbit trap -- ring -- A variation of Paul Carlson's pseudo 3d method, by Terry W. Gintz. o,o2,z,ex
5 -- Orbit trap -- square -- A variation of Paul Carlson's psuedo 3d method, by Terry W. Gintz. o,o2,z,ex
6 -- Orbit trap -- four circles -- by Paul Carlson. o,o2,z,ex
7 -- Convergence 0 -- standard (difference) -- break loop when convergent limit(cl) is reached; by Terry W. Gintz -- Exponential smoothing by Ron Barnett. r(enormalize), m Note 8
8 -- Convergence 1 -- difference of two square/adds -- break loop when convergent limit(cl) is reached; by Terry W. Gintz -- Exponential smoothing by Ron Barnett. r, m
9 -- Convergence 2 -- absolute difference -- break loop when convergent limit(cl) is reached; by Terry W. Gintz -- Exponential smoothing by Ron Barnett. r, m
10 -- Convergence 3 -- signed or -- break loop when convergent limit(cl) is reached; by Terry W. Gintz -- Exponential smoothing by Ron Barnett. r, m
11 -- Convergence 4 -- absolute or -- break loop when convergent limit(cl) is reached; by

Terry W. Gintz. Exponential smoothing by Ron Barnett. r, m
12 -- Bubble method #1 -- Bubble method for Mandelbrot/Julia sets; break loop when mod(z)>bailout -- by Paul Carlson. o
13 -- Atan method #1 -- Atan method for Mandelbrot/Julia sets; break loop when mod(z)>bailout -- by Paul Carlson. m
14 -- Level-Indexed -- Level set method from The Beauty of Fractals; break loop when mod(z)>bailout. Exponential smoothing by Ron Barnett. m
15 -- Orbit trap -- flower #1 -- by Paul Carlson. o,z
16 -- Orbit trap -- Newton rings -- A variation of Paul Carlson's pseudo 3d method, for convergent-type formulas. by Terry W. Gintz. o,o2,z,ex
17 -- Orbit trap -- Newton squares -- A variation of Paul Carlson's pseudo 3d method, for convergent-type formulas. by Terry W. Gintz. o,o2,z,ex
18 -- Orbit trap -- Newton globes -- A variation of Paul Carlson's pseudo 3d method, for convergent-type formulas. by Terry W. Gintz. o,o2,z,ex
19 -- Orbit trap -- Newton bubbles -- A variation of Paul Carlson's pseudo 3d method, for convergent-type formulas. by Terry W. Gintz. o,ex
20 -- Orbit trap -- Newton balls -- A variation of Paul Carlson's pseudo 3d method, for convergent-type formulas. by Terry W. Gintz. o,o2,z,ex

Fractal.fil is the filter library. These are similar to the built-in filters, also derived from the source code of Steven C. Ferguson. They feature improved operation in the scope of the default plug-ins. The filter library currently contains 26 filters(and a null filter for outside programs.) All filters support the magnify extension. Filters 23 and 25 also use the orbit extension.

(The notes below describe edit boxes for the default configuration dialog used with pre-v1.03 plug-ins. V1.03 plug-in actions have their own specific configuration dialogs with edit boxes which are roughly the same as those described here.)
Note 1: Newton/Halley extensions are a number 0-24 entered in the Newton arg box:
0 = relaxed -- z=z-sf/d(f).
1 = modified -- z=z-f/(d(f)+si).
2 = constant(c) -- z=z-f/(d(f)+c).
3 = relaxed/modified -- z=z-sf/(d(f)+si).
4 = premodified -- z=sz-f/d(f).
add 10 to arg for Halley's method.
add 20 to arg for Halley's Irrational method.
All Newton extensions except 2(constant) use the "s" variable entered in the Coefficient box. This is usually a small non-zero real number .1-2.0. When s=1.0, the standard Newton/Halley is plotted.
Note 2: The exponential extension is a number entered in the Exponent box.
Note 3: The magnify extension uses the Magnify 1 box for image enhancement. Start with 50 for default rendering methods and formula #33. Usually the factor is increased to the point just below where the colors start to overlap in palette use, for maximum smoothness in the image.
Note 4: The orbit extension uses the O-trap radius #1 box to set the diameter of the orbit trap(epsilon).
Note 5: The orbit2 extension uses the O-trap radius #2 box to define an optional "window"

inside the orbit trap area. This area becomes transparent to points beneath it. The window width should be less than the orbit trap diameter.
Note 6: The z-pen extension uses the z pen to map points that fall outside the orbit trap and escape the bailout.
Note 7: The exclude option uses the Exclude box to enter the iteration that orbit-trap testing begins. Points before that iteration are excluded from orbit testing.
Note 8: The renormalization option uses the Newton arg box to select an optional convergent method. If arg is 1, points are tested for convergence to 1 and infinity, before the default convergence tests are applied.

17.7 Bibliography

Bibliography

Complex Mathematics
Churchill, Ruel.V. and Brown, James Ward: "Complex Variables and Applications", Fifth Edition, McGraw-Hill Publishing Company, New York, 1990.

Korn, Granino A. and Korn, Theresa M.: "Manual of Mathematics, McGraw-Hill Publishing Company, New York, 1967.

Fractal Theory
Barnsley, Michael: "Fractals Everywhere", Academic Press, Inc., 1988.

Devaney, Robert L.: "Chaos, Fractals, and Dynamics", Addison-Westley Publishing Company, Menlo Park, California, 1990.

Mandelbrot, Benoit B.: "The Fractal Geometry of Nature", W.H.Freeman and Company, New York, 1983.

Peitgen, H.-O. and Richter, P.H.: "The Beauty of Fractals", Springer-Verlag, Berlin Heidelberg, 1986.

Formulas and Algorithms
Burington, Richard Stevens: "Handbook of Mathematical Tables and Formulas", McGraw-Hill Publishing Company, New York, 1973.

Kellison, Stephen G.: "Fundamentals of Numerical Analysis", Richard D. Irwin,Inc. Homewood, Illinois, 1975.

Peitgen, Heinz-Otto and Saupe, Deitmar: "The Science of Fractal Images", Springer-Verlag, New York, 1988.

Pickover, Clifford A.: "Computers, Pattern, Chaos and Beauty", St. Martin's Press, New York, 1990.

Stevens, Roger T.: "Fractal Programming in C", M&T Publishing, Inc.,Redwood City, California, 1989.

Wegner, Tim, Tyler,Bert, Peterson, Mark and Branderhorst, Pieter: "Fractals for Windows", Waite Group Press, Corte Madera, CA, 1992.

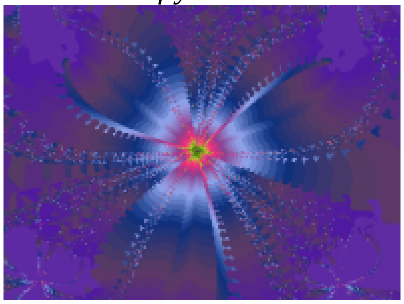
Wegner, Tim and Tyler, Bert: "Fractal Creations", Second Edition, Waite Group Press, Corte Madera, CA, 1993.

Whipkey, Kenneth L. and Whipkey, Mary Nell: "The Power of Calculus", John Wiley & Sons, New York, 1986.

17.8 About Firefly

About Firefly

>>>>> *Firefly*™ v3.038 ©1998 -- 2005 by Terry W. Gintz



Firefly graphs formulas based on 4-D complex number planes. Firefly currently supports the Mandelbrot set, Julia sets, and Phoenix curves, with millions of mapping variations. There is the choice of palette-based or generalized coloring modes. The complex math functions supported include sin(z), sinh(z), z^z, e^z, z^n, sqrt(z), cos(z), cosh(z), tan(z), tanh(z), log(z), ln(z), n^z and others, including the gamma and Legendre functions. Firefly can use any fractal plug-in that supports the Ferguson/Gintz interface spec(v1.03).

Up to two formulas for z using the above functions may be plotted, using traditional rules for generating Mandelbrot sets (Benoit B. Mandelbrot) and Julia sets (G. Julia.) Also, there are mapping options that use non-traditional methods, such as the epsilon-cross method (Clifford A. Pickover), renormalization and IFS (Michael Barnsley).

Since the formula parser is an interpreter, with its inherent lack of speed, one hundred 'popular' and unusual formulas have been hard-coded to reduce graphing time by 40 to 60 percent. Several of the built-in formulas are capable of producing graphs that the program could not generate through the parser alone, such as applying the Mandelbrot set to Newton's method for solving quadratic equations. Hypercomplex extensions (as described in Fractal Creations) have been incorporated into most of the Mandelbrot and Julia functions except for the Quaternion set (this is itself a special 4D version of the Mandelbrot set z^2+c.), and the real Newton formulas and formulas f3-f9 by Sylvie Gallet.

Firefly requires a true-color video adapter for best results. It may work in 16-bit (high color), but this hasn't been tested thoroughly.

Memory requirements for Firefly vary with the size of the drawing area Firefly opens on, ranging from approximately 3 megabytes memory for a 640X480 area to 48 megabytes for a 2048X1536 area. Special routines have been added to reduce memory requirements for large bitmaps (up to 14400X10800) by writing these directly to a file instead of using a memory bitmap. For poster size pictures (up to 96000X72000), a segmenting option is available to divide pictures into up to 225 pieces. The segments can be tiled into one large bitmap using a program such as Richard Paasen's Image Arithmetic.

Acknowledgements: many thanks to Paul Carlson for providing me his algorithms for 3D-like fractals, and allowing me to incorporate his ideas into Firefly. Also, special thanks to Ron Barnett for his help in setting up the animation routines, to Earl Hinrichs for sharing his unique programming methods on the fractal art and programmer's lists, and to Piet van Oostrum for his T2MF, without which the midi support would still be a far off dream.

Some improvements in the "intuitive feel" of Firefly are due to the timely suggestions of Sergio Capoleoni. Thanks, Sergio!

The multi-windowing interface in Firefly is courtesy of that extraordinary and prolific fractal programmer, Steven C. Ferguson, whose filters and 2D coloring routines I have also included in Firefly. Steve's contributions to the look and feel of Firefly and previous versions of my program have had a deep impact on my fractal imaging experiments.

Last but not least, remercia te très beaucoup, to Jean-Pierre Louvet, for his timely suggestions for improving ZPlot 24, Fractal and Firefly.

For a short history of this program, see Chronology.

17.8.1 Chronology

Chronology

History of this program:

In September 1989, I first had the idea for a fractal program that allowed plotting all complex functions and formulas while attending a course on College Algebra at Lane College in Eugene, Oregon. In November 1989, ZPlot 1.0 was done. This Amiga program supported up to 32 colors, 640X400 resolution, and included about 30 built-in formulas and a simple formula parser.

May 1990 -- ZPlot 1.3d -- added 3D projections for all formulas in the form of height fields.

May 1991 -- ZPlot 2.0 -- first 236-color version of ZPlot for Windows 3.0.

May 1995 -- ZPlot 3.1 -- ZPlot for Windows 3.1 -- 60 built-in formulas. Added hypercomplex support for most built-in formulas.

May 1997 -- ZPlot 24.02 -- first true color version of ZPlot -- 91 built-in formulas. Included support for 3D quaternion plots, Fractint par/frm files, Steve Ferguson's filters, anti-aliasing and Paul Carlson's orbit-trap routines.

June 1997 -- ZPlot 24.03 -- added Earl Hinrichs Torus method.

July 1997 -- ZPlot 24.08 -- added HSV filtering.

December 1997 -- Fractal Elite 1.14 -- 100 built-in formulas; added avi and midi support.

March 1998 --Split Fractal Elite into two programs, Dreamer and Medusa (multimedia.)

April 1998 -- Dofo 1.0 -- supports new Ferguson/Gintz plug-in spec.

June 1998 -- Dofo-Zon -- redesigned multi-window interface by Steve Ferguson, and includes Steve's 2D coloring methods.

August 1998 --Dofo-Zon Elite -- combination of Fractal Elite and Dofo-Zon

October 1998 -- Dofo-Zon Elite v1.07 -- added orbital fractals and IFS slide show.

November 1998 -- Dofo-Zon Elite v1.08 -- added lsystems.

April 1999 -- Split Dofo-Zon Elite into two programs: Fractal Zplot using built-in formulas and rendering methods, and Dofo-Zon to support only plug-in formulas and rendering methods.

May 1999 -- Fractal Zplot 1.18 -- added Phong highlights, color-formula mapping and random fractal methods.

June 1999 -- completed Fractal ViZion -- first version with automatic selection of variables/options for all fractal types.

July 1999 -- Fractal Zplot 1.19 -- added cubic Mandelbrot support to quaternion option; first pc fractal program to render true 3D Mandelbrots.

October 1999 -- Rebirth of Dofo-Zon Elite. Additional blending methods for 2D coloring modes, new zoom and merge options, Phong highlights and color-formula mapping for 3D fractal types, and new export options for 3D objects. Extended support for action lists to all four figures.

September 2000 -- Fractal Zplot 1.22 -- added support for full-screen AVI video, and extended quaternion design options.

October 2000 -- QuaSZ (Quaternion System Z) 1.00 -- stand alone quaternion/hypernion/cubic Mandelbrot generator

November 2000 -- Added octonion fractals to QuaSZ 1.01.

March 2001 -- Cubics 1.0 -- my first totally-3D fractal generator.

May 2001 -- QuaSZ 1.03 -- added Perlin noise and improved texture mapping so texture tracks with animations.

June 2001 -- Fractal Zplot 1.23 -- added Perlin noise and quat-trap method.

July 2001 -- QuaSZ 1.05 -- improved performance by converting many often-used dialogs to non-modal type.

October 2001 -- FraSZle 1.0, QuaSZ formula and algebra compatible version of Fractal Zplot

November 2001 -- DynaMaSZ 1.0, the world's first Dynamic Matrix Systems fractal generator

May 2002 -- DynaMaSZ SE 1.04 (unreleased version)-- scientific edition of DMZ, includes support for user-variable matrix dimensions (3X3 to 12X12)

January 2003 -- Pod ME 1.0 -- first stand-alone 3-D loxodromic generator, Hydra 1.0 -- first 3-D generator with user-defined quad types and Fractal Projector a Fractal ViZion-like version of DMZ SE limited to 3X3 matrices

May 2003 -- FraSZle and Fractal Zplot 3.052 -- added random 3D orbital fractals, new 3D export methods, upgraded most frequently-used dialogs to non-modal type and added genetic-style function type. FZ now based on FraSZle except for built-in formula list and Newton support.

June 2003 -- Firefly -- a strictly 2-D version of Dofo-Zon , complete with remote and buffered generalized coloring.

Index

- A -

- B -

- C -

- D -

- E -

- F -

- H -

- I -

- M -

- P -

- R -

- S -

- T -

- V -