fraktaler-3-0-303-gc718207

Claude Heiland-Allen

2022-06-07

Contents

Fraktaler 3

Fast deep escape time fractals.
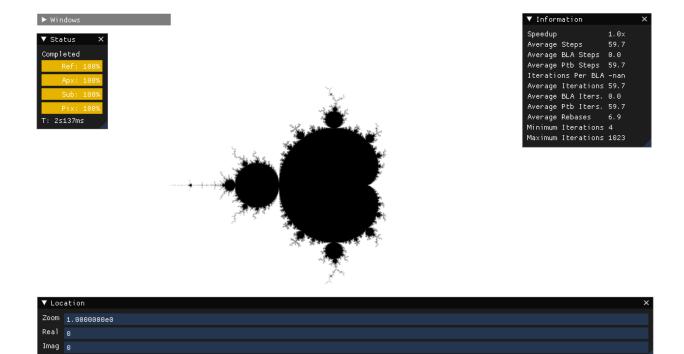
https://fraktaler.mathr.co.uk

Figure 1: Screenshot of Fraktaler 3

Live

Try Fraktaler 3 live online in your web browser.

https://fraktaler.mathr.co.uk/live/latest

Requires support for `SharedArrayBuffer`, among other web APIs. (This rules out Firefox/Fennec on Android at the moment.)

Performance is significantly slower than native versions, which are available for download below.

Download

https://fraktaler.mathr.co.uk/download

Run

Run GUI

You need support for recent OpenGL. If you don't have it, the program window may appear briefly before closing without any error messages visible.

On Microsoft Windows, if your GPU drivers do not support it you can install Mesa 3D and the Vulkan Runtime from:

- https://github.com/pal1000/mesa-dist-win/releases
- https://vulkan.lunarg.com/sdk/home#windows

Use the `mesa-dist-win` per-app deployment script.

```
./fraktaler-3-gui
```

Run CLI

```
./fraktaler-3-cli parameter.f3.toml
```

or if you have OpenCL, edit the parameter to add

```
opencl.platform = 0
opencl.device = 0
```

(replaced with the indices of your desired platform and device) and run

```
./fraktaler-3-cl parameter.f3.toml
```

OpenCL is only good until about 1e300 zoom (double precision), and progress reporting is less frequent. Throughput may be higher depending on hardware. OpenCL with CPU is typically faster than the regular CPU code. OpenCL with GPU may or may not be faster than OpenCL with CPU.

Run Web

Configure web server with headers:

```
Cross-Origin-Embedder-Policy: require-corp
Cross-Origin-Resource-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
```

Make sure `*.wasm` is served with MIME type `application/wasm`

Serve the `live/` sub-folder. Needs httpS for non-localhost domains.

You must serve the corresponding source code to comply with the license.

User Manual

Navigation

The F10 key toggles the graphical user interface windows, so you can focus on exploring the fractal. If you do not have a keyboard, you can manually close all the windows by deselecting their checkboxes in the main Fraktaler 3 window, which can also be collapsed by clicking on the small triangle in the top left.

The fractal can be explored with a mouse. Left mouse down on the desired new image center and drag to set the new image size; a rectangle is highlighted during the gesture. Release the left mouse button to confirm the new view, or press the right mouse button (while the left button is still held) to cancel the action. Alternatively use the scroll wheel to zoom in and out around the mouse cursor position. The middle mouse button centers the view on the click location.

The fractal can be explored with a keyboard. Numeric keypad keys 1-9 zoom to different quadrants of the view (1 is bottom left, 9 is top right, 4 is middle left, and so on, as per usual layout). The 0 key zooms out. Numeric keypad keys + and - adjust the maximum iteration count (doubling and halving respectively), which can also be set in the Bailout window.

The fractal can be explored with multi-touch. One touch translates the view. Two touches zoom and rotate. Three touches enables stretching or skewing the image. If you have no multi-touch device, but do have a mouse, you can use multi-touch emulation. Hold Ctrl+Shift and the left mouse button to add or move a touch point. Hold Ctrl+Shift and press the right mouse button to delete a touch point. Delete all touch points to finish the gesture and confirm the action.

Fraktaler 3 Window

This window has toggles to open/close all the other subwindows.

Input/Output Window

This has a Home button to zoom out to the original view. You must set the checkbox to the left to enable this to avoid accidents. There are also buttons to Load and Save, which can be as parameter file text (suggested extension .f3.toml) or images (EXR format, extension .exr). Clicking the Load or Save buttons opens a file browser dialog. Note: saved images do not yet include parameter metadata, so be sure to save a parameter file too if you want to return to the location later.

Formula Window

The default formula is the Mandelbrot set, with one line with |X|, |Y|, -X, -Y all unchecked and P=2. This corresponds to the familiar formula (X+iY)^2 + C. If you check both |X| and |Y| then you get the Burning Ship (|X|+i|Y|)^2 + C, if instead you check -Y you get the Mandelbar (aka Tricorn) (X-iY)^2 + C. The + button on the right lets you add more than one formula, which can be edited independently. These are iterated in an interleaved fashion, one line after the other in a loop, creating hybrid escape time fractals. Note: reference orbit processing and memory requirements increase with each line (N lines need N times the amounts total as 1 line);.

Colour Window

This window lets you adjust the colouring algorithm. Currently you can choose between monochrome (black on white) or rainbow colourings.

Status Window

Shows various progress bars to show how rendering is proceeding. There is also a timer.

Location Window

Shows the coordinates and magnification of the view.

Reference Window

Shows the coordinates and period (if any) of the reference (which is usually the image center).

Bailout Window

Adjust maximum iteration count. The first two items should usually be the same, and should be increased if there are solid regions that look out of place. The third item can be increased for complex images if increasing the first two does not fix the issue. Use the information window to diagnose the necessary iteration counts.

The escape radius is adjusted at the bottom, decrease it for high power formulas if unsightly rings appear around the fractal.

Transform Window

Adjust image transformation, including reflection (useful if your Burning Ship is upside down), rotation, and stretch. The exponential map feature is not so useful in the graphical program, but can be used in the command line version for rendering a zoom out sequence for later assembly into a video using zoomasm (https://mathr.co.uk/zoomasm).

Algorithm Window

Contains advanced algorithm tuning options. Be careful if you adjust these as sometimes bad images can result.

Information Window

Displays statistics on various aspects of the calculations, including a speedup factor for the bilinear approximation algorithm vs doing regular perturbation iterations for each pixel.

Quality Window

Control image quality. Increasing top slider decreases quality (but increases speed) by subsampling the image. Increasing the bottom slider increases quality by computing many versions of the image and averaging them. Setting the bottom slider to 0 will compute more subframes indefinitely, allowing you to stop when the quality gets high enough for you.

Newton Zooming Window

Zooms automatically to mini-sets or embedded Julia sets deep in the fractal. Set the options (each action includes the ones above), then select the activate checkbox and left-click in the image where you want to zoom. Remember to deselect the activate checkbox if you want to use the left mouse zooming feature.

Note: currently there is a bug with periodic references. For correct images, open the reference window and clear the period field.

About Window

Displays version information and software licenses.

Source

You can browse the source code repository at:

https://code.mathr.co.uk/fraktaler-3

Build

Source Dependencies

```
git clone https://github.com/ocornut/imgui.git
git clone https://github.com/AirGuanZ/imgui-filebrowser.git
git clone https://github.com/ToruNiina/toml11.git
git clone https://code.mathr.co.uk/fraktaler-3.git
```

Debian Dependencies

Bullseye or newer is required.

```
sudo apt install \
  build-essential \
  clang-11 \
  git \
  libglew-dev \
  libglm-dev \
  libmpfr-dev \
  libmpfrc++-dev \
  libomp-11-dev \
  libopenexr-dev \
  libsdl2-dev \
  p7zip \
  pkg-config \
  xxd
```

Windows Dependencies

For cross-compilation from Debian.

```
sudo dpkg --add-architecture i386
sudo apt update
sudo apt install \
  build-essential \
  git \
  mingw-w64 \
  p7zip \
  wine32 \
  wine64 \
  wine-binfmt \
  xxd
sudo update-alternatives --set x86_64-w64-mingw32-g++ /usr/bin/x86_64-w64-mingw32-g++-posix
sudo update-alternatives --set x86_64-w64-mingw32-gcc /usr/bin/x86_64-w64-mingw32-gcc-posix
sudo update-alternatives --set x86_64-w64-mingw32-gfortran /usr/bin/x86_64-w64-mingw32-gfortran-posix
sudo update-alternatives --set x86_64-w64-mingw32-gnat /usr/bin/x86_64-w64-mingw32-gnat-posix
sudo update-alternatives --set i686-w64-mingw32-g++ /usr/bin/i686-w64-mingw32-g++-posix
sudo update-alternatives --set i686-w64-mingw32-gcc /usr/bin/i686-w64-mingw32-gcc-posix
sudo update-alternatives --set i686-w64-mingw32-gfortran /usr/bin/i686-w64-mingw32-gfortran-posix
sudo update-alternatives --set i686-w64-mingw32-gnat /usr/bin/i686-w64-mingw32-gnat-posix
```

Use the `prepare.sh` script to download and build dependencies for your architecture. For help:

```
./build/prepare.sh -h
```

Windows i686

```
make SYSTEM=i686-w64-mingw32
```

Windows x86__64

```
make SYSTEM=x86_64-w64-mingw32
```

Windows armv7 You need `llvm-mingw` because `gcc-mingw` does not support Windows on ARM: https://github.com/mstorsjo/llvm-mingw

Note: `-lopengl32` is not supported upstream yet, so the GUI won't compile.

Note: Wine is untested. Microsoft Windows is untested.

```
make SYSTEM=armv7-w64-mingw32
```

Windows aarch64 You need `llvm-mingw` because `gcc-mingw` does not support Windows on ARM: https://github.com/mstorsjo/llvm-mingw

Note: `-lopengl32` is not supported upstream yet, so the GUI won't compile.

Note: Wine does not yet support `__C_specific_handler`, so it won't run in Wine. Microsoft Windows is untested.

```
make SYSTEM=aarch64-w64-mingw32
```

Emscripten Dependencies

Use the `prepare.sh` script to download and build dependencies for the `emscripten` architecture. For help:

```
./build/prepare.sh -h
```

Build For Android

Use the `android.sh` script to download and build dependencies for Android. Needs Android command line tools, SDK, NDK. Set environment variables to configure, for example:

```
ANDROID_HOME=${HOME}/opt/android
ANDROID_NDK_HOME=${ANDROID_HOME}/ndk/23.1.7779620
PATH="${ANDROID_HOME}/tools:$PATH"
PATH="${ANDROID_HOME}/platform-tools:$PATH"
PATH="${ANDROID_NDK_HOME}:$PATH"
./build/android.sh prepare
./build/android.sh
```

Default is a debug build (runs slow). Release build requires signing.

Build Documentation

Needs `pandoc`. Built as part of release.

Build Release

Builds all architectures and documentation ready for release. Does not yet include Android.

```
./build/release.sh
```

Legal

Fraktaler 3 – Fast deep escape time fractals

Copyright (C) 2021,2022 Claude Heiland-Allen

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, version 3.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see https://www.gnu.org/licenses/.

---

https://mathr.co.uk