

fraktaler-3-0-180-g145335e

Claude Heiland-Allen

2021-12-18

Contents

Fraktaler 3	1
Live	1
Download	1
Run	2
Run GUI	2
Run CLI	2
Run Web	2
Source	3
Build	3
Source Dependencies	3
Debian Dependencies	3
Windows Dependencies	3
Emscripten Dependencies	4
Build For Android	4
Build Documentation	4
Build Release	4
Legal	5

Fraktaler 3

Fast deep escape time fractals.

<https://fraktaler.mathr.co.uk>

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>

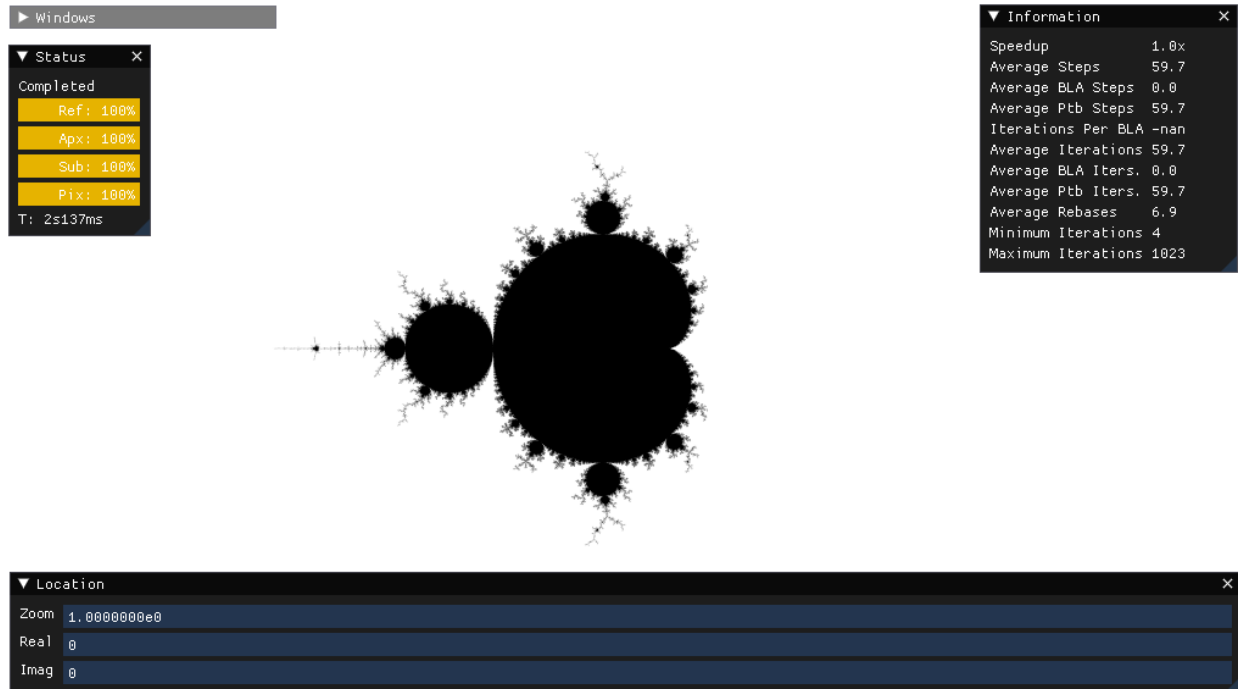


Figure 1: Screenshot of Fraktaler 3

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
```

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.

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://code.mathr.co.uk/fraktaler-3.git
```

Debian Dependencies

Bullseye recommended. Enable backports for Buster.

```
sudo apt install \
  build-essential \
  clang-11 \
  git \
  libglew-dev \
  libglm-dev \
  libmpfr-dev \
  libmpfr-c++-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:

```
./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:

```
./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"
```

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.

```
./release.sh
```

Legal

Fraktaler 3 – Fast deep escape time fractals

Copyright (C) 2021 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>