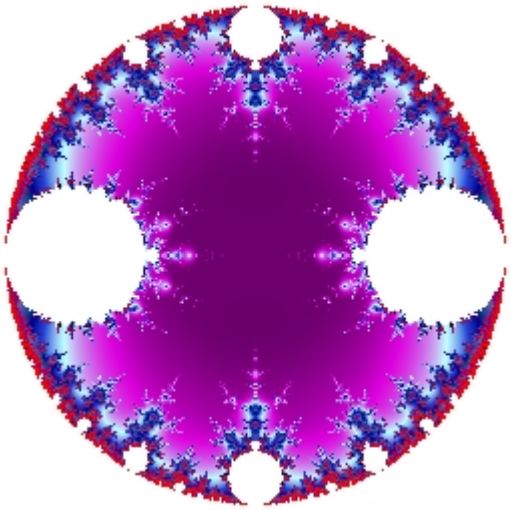
# Mini-Fractals

# Introduction





## Welcome

Greeting !

Have Fun playing with this Toy !

Show it to the Kids !

It just might get them interested in algebra.

The program is a beta, as I am not an OpenGL expert, just an intermediate.
However, I have been developing software for 3 decades now, so the application is fairly robust.

The source code, if in the installer version you used, has been put in your personal folder, might be the User Folder on Windows 7 and Vista, and/or the My Documents Folder.



## What's new

What's New? Everything about this application.

It has many added features from the original OpenCL sample it came from.

And many more features than any one sample used in it's creation.

So I guess it is a good contribution to the community.

And it's just too much fun to play with ...



## About

This Code sample application was originally a R+D project to explore GPU programming, with Fractals in

mind. The code will eventually find it's way into the 'real' project: Many-Fractals. The application was originally written with glut using OpenCL, and there is a binary executable for that included with some distros.
Email: author@manyfractals.org if you would like the code for the OpenCL sample. I'll try to post that too.

The new, and big item with this OpenGL sample is the GLSL complex (function) library. I hope this will be very helpful to other programmers.
The GLSL language does not allow C++ struct to have operators so as a result only functions for these operators can be written. To handle this, I allow Users of the executable to use operators in a text edit box, and then 'translate' them into these functions. Right now only + , - , * , / are translated. The Boolean functions aren't done yet, but the code is down to just rote repetition for these operators. However, all complex functions are fully available as valid entry in the User Enter Equations/Function, just use [] instead of () ... sorry, cheap translator.



## CPOL License

THE RIGHTS CONTAINED HEREIN IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO ACCEPT AND BE BOUND BY THE TERMS OF THIS LICENSE, YOU CANNOT MAKE ANY USE OF THE WORK.

Definitions.

"Articles" means, collectively, all articles written by Author which describes how the Source Code and Executable Files for the Work may be used by a user.

"Author" means the individual or entity that offers the Work under the terms of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works.

"Executable Files" refer to the executables, binary files, configuration and any required data files included in the Work.

"Publisher" means the provider of the website, magazine, CD-ROM, DVD or other medium from or by which the Work is obtained by You.

"Source Code" refers to the collection of source code and configuration files used to create the Executable Files.

"Standard Version" refers to such a Work if it has not been modified, or has been modified in accordance with the consent of the Author, such consent being in the full discretion of the Author.

"Work" refers to the collection of files distributed by the Publisher, including the Source Code, Executable Files, binaries, data files, documentation, whitepapers and the Articles.

"You" is you, an individual or entity wishing to use the Work and exercise your rights under this License.

Fair Use/Fair Use Rights. Nothing in this License is intended to reduce, limit, or restrict any rights arising from fair use, fair dealing, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

License Grant. Subject to the terms and conditions of this License, the Author hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

You may use the standard version of the Source Code or Executable Files in Your own applications.

You may apply bug fixes, portability fixes and other modifications obtained from the Public Domain or from the Author. A Work modified in such a way shall still be considered the standard version and will be subject to this License.

You may otherwise modify Your copy of this Work (excluding the Articles) in any way to create a Derivative Work, provided that You insert a prominent notice in each changed file stating how, when and where You changed that file.

You may distribute the standard version of the Executable Files and Source Code or Derivative Work in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution.

The Articles discussing the Work published in any form by the author may not be distributed or republished without the Author's consent. The author retains copyright to any such Articles. You may use the Executable Files and Source Code pursuant to this License but you may not repost or republish or otherwise distribute or make available the Articles, without the prior written consent of the Author.

Any subroutines or modules supplied by You and linked into the Source Code or Executable Files of this Work shall not be considered part of this Work and will not be subject to the terms of this License.

Patent License. Subject to the terms and conditions of this License, each Author hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, import, and otherwise transfer the Work.

Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You agree not to remove any of the original copyright, patent, trademark, and attribution notices and associated disclaimers that may appear in the Source Code or Executable Files.

You agree not to advertise or in any way imply that this Work is a product of Your own.

The name of the Author may not be used to endorse or promote products derived from the Work without the prior written consent of the Author.

You agree not to sell, lease, or rent any part of the Work. This does not restrict you from including the Work or any part of the Work inside a larger software distribution that itself is being sold. The Work by itself, though, cannot be sold, leased or rented.

You may distribute the Executable Files and Source Code only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy of the Executable Files or Source Code You distribute and ensure that anyone receiving such Executable Files and Source Code agrees that the terms of this License apply to such Executable Files and/or Source Code. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute the Executable Files or Source Code with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License.

You agree not to use the Work for illegal, immoral or improper purposes, or on pages containing illegal, immoral or improper material. The Work is subject to applicable export laws. You agree to comply with all such laws and regulations that may apply to the Work after Your receipt of the Work.

Representations, Warranties and Disclaimer. THIS WORK IS PROVIDED "AS IS", "WHERE IS" AND "AS AVAILABLE", WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OR GUARANTEES. YOU, THE USER, ASSUME ALL RISK IN ITS USE, INCLUDING COPYRIGHT INFRINGEMENT, PATENT INFRINGEMENT, SUITABILITY, ETC. AUTHOR EXPRESSLY DISCLAIMS ALL EXPRESS, IMPLIED OR STATUTORY WARRANTIES OR CONDITIONS, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF MERCHANTABILITY, MERCHANTABLE QUALITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY WARRANTY OF TITLE OR NON-INFRINGEMENT, OR THAT THE WORK (OR ANY PORTION THEREOF) IS CORRECT, USEFUL, BUG-FREE OR FREE OF VIRUSES. YOU MUST PASS THIS DISCLAIMER ON WHENEVER YOU DISTRIBUTE THE WORK OR DERIVATIVE WORKS.

Indemnity. You agree to defend, indemnify and hold harmless the Author and the Publisher from and against any claims, suits, losses,

damages, liabilities, costs, and expenses (including reasonable legal or attorneys' fees) resulting from or relating to any use of the Work by You.

Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL THE AUTHOR OR THE PUBLISHER BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK OR OTHERWISE, EVEN IF THE AUTHOR OR THE PUBLISHER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Termination.

This License and the rights granted hereunder will terminate automatically upon any breach by You of any term of this License. Individuals or entities who have received Derivative Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 6, 7, 8, 9, 10 and 11 will survive any termination of this License.

If You bring a copyright, trademark, patent or any other infringement claim against any contributor over infringements You claim are made by the Work, your License from such contributor to the Work ends automatically.

Subject to the above terms and conditions, this License is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, the Author reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

Publisher. The parties hereby confirm that the Publisher shall not, under any circumstances, be responsible for and shall not have any liability in respect of the subject matter of this License. The Publisher makes no warranty whatsoever in connection with the Work and shall not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. The Publisher reserves the right to cease making the Work available to You at any time without notice

Miscellaneous

This License shall be governed by the laws of the location of the head office of the Author or if the Author is an individual, the laws of location of the principal place of residence of the Author.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this License, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed herein. There are no understandings, agreements or representations with respect to the

Work not specified herein. The Author shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Author and You.

# Getting Started

Click Make Fractal Button.
Try the different Orbit Functions, clicking Make-Fractals button after each selection.
Try the Julia Check Box with Z^F+C Orbit equation. Click Zoom.
Zoom will put the App in animation mode.
After Clicking Zoom, move the mouse pointer over the image window and move it around.
Click the left mouse button to Zoom in.
Click the right mouse button to zoom out.
Double Click toggles Zoom Function.

The App uses 8 decimal digit float point numbers, so one can't magnify in very deep, but the App can make the images fast, especially User Entered Equations, as the code is fully compiled and optimized by the GLSL compiler.

The app is designed to be the most bang for the byte, and show off lot of good code examples.



## System requirements

It's hard to say the exact requirements.
The App works on ati 4670, but with strange graphic glitches, your mileage may vary.
I think it will work fine on ati 4700 series and above just fine.
It does work fine on ati 5450, and likely will on all ati 5000 series GPUs.
I think the same era nVidia's will show the same results, other than weird bugs.
The App was developed on AMD A10-5700 with ati 7660D GPU (384 shaders).
I would say, if the machine is a single core CPU, don't bother trying it.



## GLSL shader Language

See the included GLSL PDFs included with the code.

Basically, it codes just like C code, but with reduced function set.
Remember to end each statement with a semi-colon !!

Here are the links to them at OpenGL.org, if you don't have them :

http://www.opengl.org/documentation/glsl/



## User Equations

The User Entered Equation/Functions are entered in a two line high Edit Box just above the debug/info output box on the bottom of the main window.

The Code entered there is almost pure GLSL, but which version can vary. I've written it for #version 330 , 3.30
The big difference is with parenthesis '(' and ')' ... which in the language (and C/C++) has more than one use. For our User Equations, they only mean the algebraic use. Use Square brackets for all other use. Therefore to call the sine of a number use: sin[...]
The shader file has quite a few coding helpers built into it. The most helpful for our Fractal context is the Complex Functions and variables.
The Complex type is a C struct. It's members are r for real and i for imaginary.
The editor allows for addition, subtraction, multiplication and division to be written algebraically, although the GLSL language has no operators. This is done by translating the edit box code into Complex Number Functions: ADD, SUB, MUL and DIV. The last thing the translator does is translate all square brackets to parenthesis. In the translation the original parenthesis are removed. However, any original square bracket are retained and replaced with () ... so it all works out, if you simply just use [] for all non-algebraic () use.

The Complex number functions available are as follows:

re[] returns the real component
im[] returns the imaginary component
ir[] returns Complex(i,r)
abs[] absolute of complex
norm[]
arg[]
sgn[]
rec[]
pol[]
NEG[] - operator
COMPLIMENT[] ~ operator
ADD[] + operator translated for you
SUB[a,b] a - b operator translated for you
MUL[] * operator translated for you
DIV[a,b] a / b operator translated for you
pow[a,b] a^b
sqrt[]
ISEQU[a,b] is equal == operator
ISNOTEQU[a,b] != operator
ISLESS[a,b] a < b operator
ISMORE[a,b] a > b operator
ISLESSEQU[a,b] a <= b operator
ISMOREEQU[a,b] a >= b operator
exp[]
ln[]
log[] same as ln[]
log10[]
sin[]
cos[]
tan[]
sinh[]
cosh[]
asin[]
acos[]
atan[]
tanh[]
asinh[]
acosh[]
atanh[]

There are helper functions for other variables, like float, but not Complex :
MUL_ADD[a,b,c] which is ( ((a) * (b)) + (c) )
ATAN2[a,b]

hypot[r,i]
fabs[x]

Constants:

| | |
|---|---|
| HALFPI | 1.5707963f |
| LOG10 | 2.3025851f |
| LOG2 | 0.6931472f |
| e | 2.7182818f |
| pi | 3.1415927f |
| DBL_EPSILON | 2.2204460e-8 |

I, J, _I, _J
defined as :

```
I.r =  0.0 ;  I.i = 1.0 ;                  // root of z^2+1 = 0
J.r = -0.5 ;  J.i =0.5*sqrt(3.0) ;         // root of z^2+z+1 = 0
_I = DIV(1.0 , I);
_J = DIV(1.0 , J);
//     J = -0.5 , 0.8660254037844386
//    _J = -0.5 , -0.8660254037844386
//    _i = 0 , -1
//     i = 0 , 1
```

// ToDo: these need Complex-Function counter-parts to have them work with Complex ...

```
#define SWAP(a,b,t) ((t)=(a),(a)=(b),(b)=(t))
#define SIGN(a)  ((a)<0 ? -1 : 1)
#define ABS(a)   ((a)<0 ? -(a) : (a))
#define SQR(a)   ((a)*(a))
#define MIN(a,b) ((a)<(b) ? (a) : (b))
#define MAX(a,b) ((a)<(b) ? (b) : (a))
#define MIN3(a,b,c) ((a)<(b) ? ((a)<(c) ? (a) : (c)) : ((b)<(c) ? (b) : (c)))
#define MAX3(a,b,c) ((a)>(b) ? ((a)>(c) ? (a) : (c)) : ((b)>(c) ? (b) : (c)))
```

// These are Read-Only 'Input' values to the shaders.

```
uniform vec2 param;
uniform int iters;
uniform int ipow_f;    // integer power F for the edit-spin box on control-panel dialog
uniform int OrbitFunc;
uniform bool UseEsmooth;
uniform float floatG;    // our user input floating point value
uniform float fBailout;   // the float of the compare value. Note: norm(ZZ) not ABS(ZZ)
uniform vec2 viewPos;
uniform float viewZoom;
uniform float viewRatio;
uniform float anim;
uniform bool juliaMode;
uniform bool blinkMode;
```

The actual code of the User Equation in the shader file :

```
////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
//    USER ENTERED EQUATION HERE !!
//
```

```
                case 3:
                                // UseEsmooth  floatG  fBailout
                                i=0;
                                while( i < iters )
                                {
                                        dNorm = norm(ZZ);
                                        if( dNorm >= _Type1_(fBailout) ) break;
                                        if( UseEsmooth ) fMU.x += Fast_eSmooth(dNorm);
                                        i++ ;

                                        /////////////// DO NOT EDIT FOLLOWING  /////////////////

                                          // FGZ_USER_FUNCTION_INSERTION_POINT

                                        ///////////////// END /////////////////
                                        // ZZ = ADD( MUL(ZZ, ZZ), DIV( CC, ADD(_Type1_(1.0), MUL
(floatG, ZZ) ))) ;
                                        // /////
                                }
                                if( !UseEsmooth ) {
                                   fMU.x = ZZ.r ;
                                   fMU.y = ZZ.i ;
                                }

                                break;


                ////////////////////////////////////////////////////////////////////////////////
                ////////////////////////////////////////////////////////////////////////////////
```



## User Equ/Func tips

Another slight glitch in the translator :

sin[ ZZ * ipow_f ]  will cause compiler error

use:

sin[ (ZZ * ipow_f) ]  this will compile fine ... sorry

///////////////

If you get a compile error about a '}' , it's because of something else you entered or the translator botched.

//////

if you get link errors, it's all over with, the application will not work with the GL on your system.

//////

there in a known major graphics glitch on ati 4670 and ati 5450 both with 512 MB VRAM.
The issue may be too little VRAM. The author doesn't know.

The application does produce perfect graphic on ati 7660D.
A yellow square means your GPU doesn't have the right shaders for the application.

//////



## Julia Set tips

Try magnifying in first on the Mandelbrot (or Mandel like) Set, first, just a little ways, not too far. Then switch to Julia mode on the same Orbit function, and Keep on Zooming !

Enjoy !

... on some functions Julia mode doesn't work well. Keep experimenting.