

A SHORT INTRODUCTION INTO FUNKY3D

1) WHAT IT IS

Funky3D is a small application showing functions in the three-dimensional space. Functions have the common mathematical form of $z=f(x,y)$. Providing the appropriate libraries, Funky3D should compile and run on Linux, MacOS 10.x and Windows systems, using different ansi-compliant-compilers. The program is based on OpenGL using the GLUT-library. The Windows-Version is shipped with the adequate DLL. The current version directly supports the free available Windows compilers LCC-Win32 and OpenWatcom. If recompiling for yourself, you need to take care, that the GLUT-libraries are installed properly. This is important for LCC-Win32, as well as for OpenWatcom or the different Unix/Linux operating systems.

2) USAGE

The usage of Funky3D is quite straightforward. Or atleast it should be :-) Mostly you have to consider, that there are two windows. The first is dedicated to graphic-output or choosing things like zooming/scaling only, the second is a console-window for messaging and the input of functions. That means, you have to activate the right window via mouse-click. This is especially important for expression-input. A hint: Arrange the two windows on your desktop in a manner, that allows you to watch both simultaneously. To get a quick overview over Funky3D-capabilities, press D for a little demonstration. F1 shows you the usage of the most important mouse- and keyboard-commands.

3) FUNCTION INPUT/PARSER

The parser is relatively clever. For example it handles inputs like:

$2*x^2+4*x+3^y$	„full“ notation
$2x^2+4x+3^y$	more convenient and shorter
$3.1415e1x^2+xy$	scientific notation
$\sin(\pi*x)+y$	common functions and the constant Pi
$\sqrt{\text{abs}(x)}-y$	more common functions
$\text{euler}^x+\text{euler}^{\text{euler}(x+y)}$	the Euler constant 2.7181...

Error-handling is done for zero-divisions or unknown expressions. Anyway you can input and draw functions like $1/x$ or $1/(x+y)$ without getting Funky-3D confused of the infinite z-values. In the opposite you will get an divided by zero-message for explicate expressions like $x+(y/0)$ or $1/0$.

4) KNOWN BUGS/LIMITATIONS

At least on my configuration Funky3D seems to be quite stable. It only crashes in some cases, when the calculated z-values are in the region of $10e9$ or more. Possibly the tessellation under Windows-specific OpenGL fails to handle those „huge“ values. I was simply to lazy to cure this. The not documented, but working „F“-key toggles to full-screen mode. The Windows-implementation of OpenGL/GLUT may result in some little viewing-errors.

When compiling with OpenWatcom, the calling convention MUST be stack-based. This is not a matter of Funky3D. The GLUT-libraries seems not to be designed for register-based function calls.

The parser recognises the most common functions. If somebody wants more exotic functions, it is fairly easy to implement them in a copy and paste-manner. Even without full understanding, how and what the parser really does, this should be manageable. Just build in an additional „slot“ with the new function desired. Adopt the string-lengths, fill in the new function and proceed.

5) POSSIBLE IMPROVEMENTS FOR Version 1.0

Even comparing with commercial products, Funky3D does quite a good job. In some way it is nicer, more flexible and interactive than some older or even newer mathematical-packages. I don't want to

mention them here, because this may be just a matter of personal taste. Anyway, with Funky3D you get the chance to do all the things you want managed by yourself.

Of course there is room for improvements. It may be worth, to clean up the drawing-routines for the info-window. After I cracked the OpenGL-nut of how to get the real-world-coordinates of a mouse-click, I lost patience. Removing some „magic-constants“ with a general approach may give you a more usefull little „click-box-library“.

Using precompiled display-lists instead of the triangle/quads/point-routines may speed up calculations considerably. In the Demo/Benchmark-section I added an example-routine for text-output. This is not only intended to be a optical gimmick. It shows, how display lists are working in general.

Reordering the parser-atoms according to their possibility may result in a few-speed-points.

Adding a third function to be evaluated/shown would provide you with additional capabilities. Using the measuring-cross you will be able to solve any 3-dimension-equation-system with 3 variables in a graphical manner. Or maybe not, when there is no reasonable mathematical solution :-)

And of course, the info-window could be nicer. And, and ...

6) NOTICE

Funky3D was mainly developed with LCC-Win32. For non-commercial usage, this well supported C-compiler is freely available. In some aspects this C-compiler offers newer Ansi-, and/or C99-compliance then products of a well known and huge US-based company. Many thanks to the author Jacob Navia. Downloads are available under <http://www.cs.virginia.edu/~lcc-win32/>

7) HINTS FOR SUCCESSFULL COMPILING

Using LCC-Win32 you may face some troubles when compiling/running. Something seems to go wrong with the GLUT-libraries and/or the corresponding DLL's provided by Jacob Navia. At least in the distributions up to now. To overcome this potential problems, get a full set of newer GLUT-library. As a required minimum you have to have: ALL files according to the same version number. To be more specific you will need at least:

GLUT.H
GLUT32.LIB
GLUT32.DLL

Appropriate links you will find e.g. under the official OpenGL-Site <http://www.opengl.org> . For successful compiling under LCC-Win32 the following steps may be needed:

Convert ALL downloaded library-files to LCC-format and replace ALL the originally delivered glut.h, glut.lib, glut32.lib, glut.dll and glut32.dll files with the newer ones. This seems to work for the GLUT-versions 3.6 and 3.761. The later one is a little bit „hacked“, a little bigger in size, but preferable for some reasons.

Now you should be ready to compile/execute every OpenGL/GLUT-stuff like Funky3D in a click-and-go manner. If you are distributing a Funky3D-executable or a Funky3D-Clone, you should to take care to distribute the appropriate DLL, for which it was originally compiled for. Choosing this way, you should be on the secure-side.

The well-known DLL/library-„hell“ is in some sort not solely to LCC-Win32. Compiling for OpenWatcom you have to care for the appropriate libraries as well. On Linux-platforms you may face similar problems when compiling with GNU-C, moreover this compiler is providing you with a wide variety of possible target-platforms. Funky3D itself is strictly ansi-compliant, therefore you should have no severe problems compiling it for mainframes like the IBM-Z-series as well as for small devices like palmtops, as far as the target-devices support OpenGL/GLUT and in case of doubt you are providing the neccessary libraries.

For LCC-Win32-users interested in OpenGL/GLUT-stuff like Funky3D, it would possibly a great help, if Jacob Navia would spend the time, to build in the latest libraries in his standard distribution of LCC-Win32.

8) NEW IN VERSION 1.1

- As Funky3D is growing bigger and bigger (arround 2.000 lines of code now) , the parser is not simply „included“ any longer. In a more „C-project“-style the parser is an own module now, which has to be compiled and linked in the usual manner.
- In terms of documentation and readability the source-code was improved. At least I hope so.
- The info-window introduces input for a third polynom now.
- The latter allowed the implementation of an equation-solver. This solves three equations in the three-dimensional space now. The solver is based on a genetical algorithm using an adaptive step-width and is therefore relatively clever, even it is relatively simple so far. Most of the numerical solvings are reached within a few hundred iterations, if the „orientation-cross“ is set to reasonable values. If the solver fails to find a possibly correct solution, you will get an appropriate message in the info-window.
- OpenGL display lists are fully implemented for function-redrawing now. This saves considerable computing-time and should make Funky3D more useable for „older“ hardware or software-emulations for OpenGL.
- The demo-mode and the F1-help-messages had been adopted to the new capabilities. Check it out.
- Some minor changes in the parser and main-module to speed up calculations or to improve error handling.

9) LEGAL

You are free to use Funky3D and/or modify the source code for private or educational purposes. You are NOT allowed to use the program, the source code or main-portions of it for any commercial purposes without the explicit permission of the author. If you are producing „sons“ or „daughters“ of Funky3D, you should mention somehow „Funky3D“ and the original author. Have fun. I hope, this is usefull to somebody. Heinz van Saanen, Vienna, July 2003. The exact terms:

```

/* Funky3D
*
* Copyright (C) 1991, 1992, 2002, 2003 Heinz van Saanen
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or (at
* your option) any later version.
*
* 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
* General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
*/

```