

OpenGL Installation: Windows

NOTE: You might find it helpful to do a web search on Visual C++ and freeglut to find step by step instructions for a particular version of VC++. Each version of VC++ seems to move around some standard files and change the menus.

GLEW: <http://glew.sourceforge.net/>

freetglut: <http://www.transmissionzero.co.uk/software/freetglut-devel/>

Installing OpenGL for Visual Studio

1. Identify the Visual Studio folder for the version you are using. This is generally at *Program Files/Microsoft Visual Studio <version>/VC* or *Program Files (x86)/Microsoft Visual Studio <version>/VC*. Where <version> is
 - a. VS2005: 8.0
 - b. VS2008: 9.0
 - c. VS2010: 10.0
 - d. VS2012: 11.0
 - e. VS2013: 12.0
 - f. VS2015: 14.0
2. Download both GLEW and freeglut from the Web. All are available as either source or Windows binaries. If you use VC, you only need the binaries. Extract each and
 - a. Put all .dll files in *Program Files/Microsoft Visual Studio <version>/VC/bin*. Normally choose the default non-64-bit versions.
 - b. Put all .h files in *Program Files/Microsoft Visual Studio <version>/VC/include/GL*. Normally choose the default non-64-bit versions.
 - c. Put all .lib files in *Program Files/Microsoft Visual Studio <version>/VC/lib*. Normally choose the default non-64-bit versions.
3. The following files should all come with Windows. You should first have updated your graphics driver to ensure that you have recent versions
 - a. Opengl32.dll
 - b. Opengl32.lib
 - c. gl.h
4. Later versions of Visual Studio don't let you use fopen (the want you to use fopen_s, but this doesn't work for Mac). To override this:
 - a. Right click on your project and choose "Properties"
 - b. Under "Configuration Properties" go to the "C/C++" area then choose "Advanced"
 - c. On the right side in the "Disable Specific Warnings" area type "4996" (without the quotes)

Making an OpenGL Project in VS

1. You want to create an empty application. On older versions of VC++, you create a console application
2. Under project properties, click on linker and then on inputs.
 - a. Add glew32.lib and freeglut.lib to the end of the Additional Dependencies list. Don't forget to put a ; at the end of each name.

OpenGL Installation: Mac

Use Xcode to build your OpenGL projects. The current version is 5.1.1

To make a new OpenGL project do the following:

1. Create a new project
 - a. File→New→Project
 - b. Choose OS X→Application→Command Line Tool
2. Delete the default main.cpp file and drag the files you want into the project
3. Link OpenGL and GLUT with your project
 - a. Click on your project (on the left) and then select the *Built Phases* tab in the middle area
 - b. Under *Link Binary With Libraries* click the "+" button and select OpenGL and GLUT frameworks
 - i. NOTE: OSX now considers most of the GLUT functions depreciated. For the purpose of having cross-platform code we'll keep using GLUT on OSX.
4. Observe where your builds will go
 - a. By default the will go to some weird location. Go to Preferences then choose the Location Tab. The *Derived Data* area says where your build will go
 - i. Consider changing this to *Relative*
5. Build your project
 - a. Make sure you place your shader files in the same directory as the binary that is built.

OpenGL Installation: Linux

Since each flavor of Linux is a bit different your best bet is to search for solutions that will work best for your system. However below are few links to get you started:

- http://www.cs.unm.edu/~angel/BOOK/INTERACTIVE_COMPUTER_GRAPHICS/SIXTH_EDITION/linux
- <https://github.com/lkarydas/objview>

In particular, you'll likely need to install a few libraries if they aren't already on your system. Those libraries include:

- GL
- GLU
- GLUT
- GLEW

I have also provided a makefile to get you started. This makefile expects a particular file structure, in particular that the necessary include files (mat.h, Angel.h, vec.h, etc..) are in ../../include and that the InitShader code is in ../../common/

In addition, since there are so many different flavors of Linux, if you want to develop on your own Linux machine, you must make sure that it compiles and runs on tux. Tux already has all the necessary libraries installed, you just need to enable X11 forwarding to be able to see the graphics.

NOTE: The graphics cards on the tux cluster requires shader version 130 (as opposed to 150 provided). To get this to work all you need to do is change #version 150 to #version 130 in each of the *.glsl files.