

## Installing OpenGL and GLUT On Windows

### 1. Installing OpenGL

OpenGL comes preinstalled with Windows 95/98/NT/2000. In your system directory (C:\windows\system on Windows 95/98, C:\WINNT\system32\ on Windows NT/2000) you will have the files opengl32.dll and glu32.dll.

In most cases, you will also have a 3D graphics card that supports OpenGL in hardware. If not, we recommend that you buy one. All recent graphics cards support OpenGL in hardware. Please make sure that you download the latest OpenGL drivers for your graphics card from the card or chip vendors web site. You may also use glSetup, available at <http://www.glsetup.com/>, to automatically get the latest drivers for your system.

### 2. Installing GLUT

There is a Windows 95/98/NT/2000 implementation of GLUT (the OpenGL Utility Toolkit) available at <http://www.xmission.com/~nate/glut.html> and on the class web-site. To install Glut on your system the following files should be moved (or copied) to the following directories:

- Copy glut32.dll to the C:\windows\system directory (or C:\winnt\system32 if you are working on NT/2000).
- Copy glut32.lib to C:\Program Files\Microsoft Visual Studio\VC98\lib\ or to the equivalent lib directory where you have Visual C++ installed. Borland users should copy this lib-file to the lib directory of the Borland compiler.
- Copy the file glut.h to C:\Program Files\Microsoft Visual Studio\VC98\include\GL\ (or the corresponding Borland include\GL directory).

Restart windows (very important) to enable the OS to load the glut32.dll and you should be all set.

Send us email if you do not get our code to run properly on your system and be as specific about the errors you are getting as you can be.

### 3. Compiling Your OpenGL Code in Visual C++

#### *3.1. Specifying Link Libraries in Visual C++*

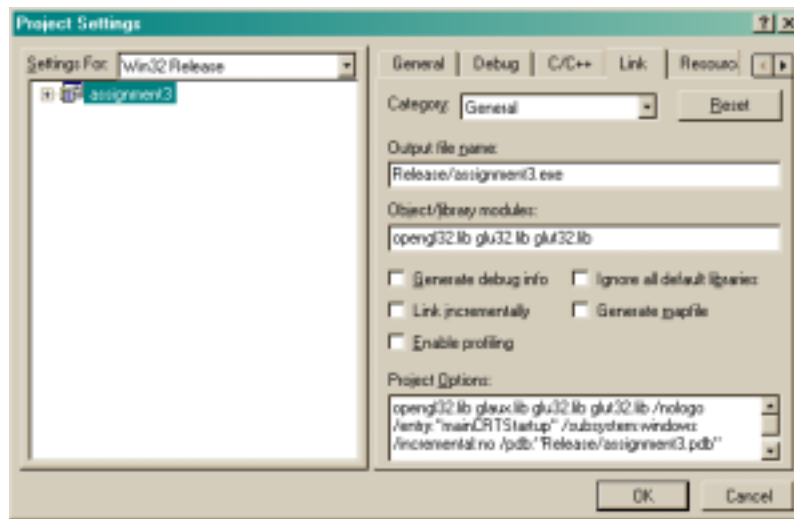
To compile an OpenGL program in Visual C++ requires that you tell the linker where to find the OpenGL and GLUT library files. Start a new project as described in the "How to

Create a Working Visual C++ Project” handout<sup>1</sup>. To shortcut things, you may also copy an existing .c/.cpp file into a project directory and double click on it to start Visual C++. Then go to Build→Build to compile the file. Visual C++ will ask you if it should create a default workspace. Click “OK”. The file will compile, but you will likely get many link errors.

You need to tell Visual C++ where to find the OpenGL libraries. Click "Settings" under the "Project" menu. Open up the Link tab. Now in the "Object/library Files" box, put in this list of .lib files:

opengl32.lib glu32.lib glut32.lib

Your project settings should look like this:



Build your code again (F7). Now you should not get any link errors and the code should compile and execute (CTRL-F5).

### 3.2. Adding New Source Code Files

If you want to start with a new file, use Open File-New and pick "C++ Source File" and name it whatever you like (for example, main.c). Look around for a FileView tab, it should be on the sidebar on your left. If you can't find it, hit Alt+0 (Alt Zero) to bring it up. Double click the "main.c" file. Then add to the top of this file the lines:

```
// Standard includes
#include <stdlib.h>
#include <stdio.h>
#include <string.h> // string-related functions
#include <time.h> // time-related functions
#include <math.h> // math functions (sin, cos, etc.)

// OpenGL includes
#include <GL/glut.h>
```

<sup>1</sup> Available on the course web page.

```
#include <GL/gl.h>  
#include <GL/glu.h>
```

Important: include <GL/glut.h> before all the other GL include files! It shouldn't matter, but under Windows it apparently does.

## 4. Debugging OpenGL

OpenGL and Windows are not always performing in the most desirable way. Our experience has shown that sometimes the behavior of an application under windows that relies on OpenGL does not crash, but at the same time does not perform the desired actions. Although you should always be suspicious that you might have a major bug in your code, restarting window can sometime help to solve the problem. At least then you know that the memory is cleared and initialized, so if the problem persists it is a bug!

OpenGL is not much fun to debug. Most calls to the API go through several layers of driver code and Visual C++ is not able to tell you where exactly your program crashes. Fortunately, the nice people at SGI have provided **gldebug**, a small application that catches all calls to OpenGL. You can download **gldebug** for Windows from the course web site.