

Visual C++ Tutorial
CSE 167, Winter 2003
Farhana Bandukwala

One of the best ways to begin understanding the structure and style of a typical OpenGL program is to enter, compile, and run a small example. The listing below contains a simple program which uses OpenGL Utility Toolkit (GLUT) to open a window. If this is the first time you've seen this sort of program, it probably will not make a lot of sense initially. Don't worry about that. For now, the goal is to use the Visual C++ environment to create, compile and execute this simple program.

```
//simplewindow.cpp
#include <windows.h>
#include <gl\glut.h>

// Called to draw scene
void renderScene(void)
{
    // Clear the window with current clearing color
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    //drawing code here

    glFlush();
}

void initState(void)
{
    glClearColor(0.0f,0.0f,0.0f,1.0f);
}

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow("Simple Window");
    glutDisplayFunc(renderScene);

    initState();

    glutMainLoop();
}
```

```
}
```

This small program does three things. First it initializes the glut library, and creates a window. Next, it sets up the display callback (renderScene) and last it initiates the main event loop (glutMainLoop). We will discuss the details of this program structure in subsequent classes. For now, this helps you set up the Visual C++ environment and get a piece of OpenGL code compiled.

Tutorial Steps:

1. Start VC++ by double clicking on its icon in the Visual C++ group of the Program Manager.
2. If VC++ has been used before on this machine, it is possible for it to come up in several different states because VC++ remembers and automatically reopens the project and files in use the last time it exited. What we want right now is a state where it has no project or code loaded. If the program starts with a dialog that says it was unable to find a certain file, clear the dialog by clicking the "No" button.
3. Go to the **Window** menu and select the **Close All** option if it is available. Go to the **File** menu and select the **Close** option if it is available to close any remaining windows. Now you are at the proper starting point .

What you see now is "normal". Along the top is the menu bar and several toolbars. Along the bottom is a status window where various messages will be displayed.

4. To create a new project for simplewindow.cpp, choose the **New** option in the **File** menu. Under the **Projects** tab, highlight **Win32 Console Application**. In the Location field type an appropriate path name or click the Browse button. Type the word "simplewindow" in for the project name, and you will see that word echoed in the Location field as well. Click the OK button.
5. In the next window, use the default selection "An empty project", click "Finish", then click "OK" once more in the next window.
6. The area along the left side of the screen will now change so that three tabs are available. The InfoView tab is still there, but there is now also a ClassView and a FileView tab. The ClassView tab

- will show you a list of all of the classes in your application and the FileView tab gives you a list of all of the files in the project.
7. Now it is time to type in the code for the program. In the **File** menu select the **New** option to create a new editor window. In the dialog that appears, make sure the **Files** tab is active and request a "C++ Source File". Make sure the "Add to Project" option is checked for Project "simplewindow", and enter "simplewindow" for "File name".
 8. Visual C++ comes with its own intelligent C++ editor, and you will use it to enter the program shown above. Type the code in the listing into the editor window. You will find that the editor automatically colors different pieces of text such as comments, key words, string literals, and so on.
 9. After you have finished entering the code, save the file by selecting the **Save** option in the **File** menu. Save it to a file named simplewindow.cpp in the new directory Visual C++ created.

In the area on the left side of the screen, click the FileView tab and expand the tree on the icon labeled "simplewindow files", then expand the tree on the folder icon labeled "Source Files". You will see the file named simplewindow.cpp. You can remove a file from a project at any time by going to the FileView, clicking the file, and pressing the delete button.

Finally, you must now tell the project to use the glut library. If you omit this step the project will not link properly, and the error messages that the linker produces will not help one bit. First, locate the glut32.lib file on your computer using the "search" option in the Start Menu (Windows Toolbar, not within Visual C++). Normally it is in C:\Program Files\Microsoft Visual Studio\VC98\Lib. If the file is in a different location, copy it to the above location. If you don't find this file, you need to contact one of the TAs.

10. Choose the **Settings** option in the **Project** menu. Make sure that the **General** tab is selected in the tab at the top of the dialog that appears. Select the tab labeled "Link". In the top left corner, select "All Configurations" from the pulldown menu. At the end of the "Object/library modules:" list, add a space and type "glut32.lib" (without quotes). Then click the OK button.

11. Having created the project file and adjusted the settings, you are ready to compile the `simplewindow.cpp` program. In the **Build** menu you will find three different compile options:

1. Compile `simplewindow.cpp` (only available if the text window for `simplewindow.cpp` has focus)
2. Build `simplewindow.exe`
3. Rebuild All

The first option simply compiles the source file listed and forms the object file for it. This option does not perform a link, so it is useful only for quickly compiling a file to check for errors. The second option compiles all of the source files in the project that have been modified since the last build, and then links them to form an executable. The third option recompiles all of the source files in the project and relinks them. It is a "compile and link from scratch" option that is useful after you change certain compiler options or move to a different platform.

12. In this case, choose the **Build simplewindow.exe** option in the **Build** menu to compile and link the code. Visual C++ will create a new subdirectory named `Debug` and place the executable named `simplewindow.exe` in that new subdirectory. This subdirectory holds all disposable (easily recreated) files generated by the compiler, so you can delete this directory when you run short on disk space without fear of losing anything important.
13. If you see compiler errors, simply double click on the error message in the output window. The editor will take you to that error. Compare your code against the code above and fix the problem. If you see a mass of linker errors, it probably means that you specified the project type incorrectly in the dialog used to create the project. You may want to simply delete your new directory and recreate it again following the instructions given above exactly.
14. To execute the program, choose the **Execute simplewindow.exe** option in the **Build** menu. A window appears. Note that the program is complete. You can move the window, re-size it, minimize it, and cover and uncover it with other windows.
15. To terminate the program, click on its system menu (the small box to the left of the title bar) and select the `Close` option.

