OpenGL is an Application Programming Interface (API) for hardware accelerated 2D and 3D graphics. **GLUT** is the Graphics Library Utility Toolkit, which adds non-graphics functionality such as controlling windows and managing system events.

These instructions describe how to get an OpenGL/GLUT project running in three C++ programming environments:

- Visual Studio 10 on Windows 7
- Code::Blocks using the GNU GCC compiler on Windows 7
- XCode on Apple OSX

Installation:

OpenGL is installed with any of the three compilers. GLUT is installed with XCode on Apple OSX.

You will need to manually install GLUT if you are using Windows.

Installing GLUT for Windows:

1. Download GLUT for Windows
    a. Go to: **http://www.xmission.com/~nate/glut.html**.
    b. Download and unzip **glut-3.7.6-bin.zip**. You should find five files in the archive.
2. Copy **glut32.dll** into a Windows system directory. This is the dynamic-link library that must be available when you run your program.
    a. Copy **glut32.dll** into **C:\Windows\system** is you're using 32-bit Windows.
    b. Copy **glut32.dll** into **C:\Windows\SysWOW64** if you're using 64-bit Windows.
    c. If you cannot copy into the system directory for some reason, you can also place the DLL into your project directory. This is less convenient since you'll have to copy the DLL each time you make a new project.
3. Install the include files and libraries somewhere accessible.
    a. Create a base folder for the GLUT files. For example, you could create **C:\mylibs\opengl**. The rest of these instructions will use **<OpenGLpath>** as a substitute for whatever base folder you've created.
    b. Copy **glut.h** from the GLUT archive to **<OpenGLpath>/include/GL**.
    c. Copy **glut32.lib** and **glut.def** from the GLUT archive to **<OpenGLpath>/lib**.

Note: You can also install GLUT directly into the compiler's directories. If you do this, you can probably avoid the project configuration steps that follow, but you'll need to be careful not to corrupt your compiler installation. This is probably not an option if you are using a university managed machine.

Project Configuration:

For each project you create, you'll need to tell you compiler and linker where to find the libraries and include files.

Visual Studio Project Configuration:

1. Select "File" → "New"→ "Project". You'll get the new project pop-up dialog.
    a. Create a "Win32 Console Application".
    b. Select "Empty Project" and create the project.
2. Tell compiler where to find GLUT include files:
    a. Open Project Properties
    b. Expand "Configuration Properties
    c. Select "C/C++"

   d. Edit "Additional Include Directories" and add **<OpenGLpath>\include**. Substitute the directory that you created when you installed GLUT for **<OpenGLpath>**.
3. Tell the linker where to find GLUT library files:
   a. Expand "Configuration Properties
   b. Select "Linker"
   c. Edit "Additional Library Directories" and add **<OpenGLpath>\lib**.
4. Add your C++ program code to the project.

Code::Blocks Project Configuration:

Code::Blocks will configure OpenGL and GLUT when you create the project.

1. Select "File" → "New"→ "Project". You'll get the new project pop-up dialog.
2. Select "GLUT Project" and hit "Go".
3. Enter the project title and Folder as usual. Hit "Next".
4. You'll be asked for GLUT's location. This should be the folder **<OpenGLpath>**, that you created when you installed GLUT.
5. Select the GNU GCC compiler and hit "Finish".
6. Add your C++ program code to the project.

Code::Blocks will give you a main.cpp that can be used to test OpenGL and GLUT. It probably will not compile. If you want to run this test program on Windows, Modify the block of code at the top of main.cpp as follows:

```
#ifdef __APPLE__
#include
<GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
```

→

```
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <windows.h>
#include <GL/glut.h>
#endif
```

XCode Project Configuration:

1. Create an empty project.
2. Create a new target, which should be a "Carbon Application".
3. Add the following frameworks to the target:
   a. GLUT.framework
   b. OpenGL.framework
4. Add your C++ program code to the project.

Additional References:

GLUT homepage:

- http://www.opengl.org/resources/libraries/glut/.

The XCode instructions are based on the following notes:

- http://cit.cs.dixie.edu/cs/3600/xcode-glut/index.html
- http://blog.onesadcookie.com/2007/12/xcodeglut-tutorial.html.