

Instructions to Set Up OpenGL Environment

Contents

1	Introduction	2
2	Set up on Windows	3
3	Set up on Linux	5
4	Set up on Mac	6
5	Test Code	7

1 Introduction

As a popular graphic API, OpenGL probably has been installed on your computer already no matter what kind of operating system you're using. But in most cases, it's not enough to use only OpenGL to achieve our goals. For example, if you want to build up a simple window under Windows, it may be a hard task for those who never do it directly without any library supported. In this course, we will take the OpenGL Utility Toolkit (a.k.a GLUT) library as a powerful tool to help us in the process of learning OpenGL. We may use other tools (such as SOIL, i.e. Simple OpenGL Image Library, to load images) which will be introduced within the corresponding exercises.

In this instruction, we will try to make it easier to set up the environment we'll use in this course. We hope you can set up the environment all by yourself following the instruction. Of course, if you have any problem, don't hesitate to contact the TA :)

2 Set up on Windows

It's useful to use an IDE to organize the program when the project we're working on becomes larger and larger. Here we suggest *Microsoft Visual Studio 2010* as the programming IDE.

OpenGL should already exist in your computer. If you're not sure, check out the following directory:

C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\include\gl

If you can find the *gl* directory with the *gl.h* and *glu.h* files in it, you can continue to the following steps.

Now what we need to do is install the GLUT library. For convenience, we have upload the library package (glutdlls37beta.zip) to the Web-Learning. There are 5 files in the compressed pack:

- glut.h
- glut.lib
- glut.dll
- glut32.lib
- glut32.dll

Follow the next steps to set up GLUT:

Step 1 Put the *glut.h* in the *V7.0A\gl* directory;

Step 2 Put the *glut.lib* and *glut32.lib* in the *V7.0A\lib* directory;

Step 3 Put the *glut.dll* and *glut32.dll* in the *C:\Windows\System32*.

Until now, the environment we need should already be set up. Try a simple program at the end of the instruction and test whether it works.

In fact, the *lib* directory mentioned above is in the VS2010 default library directories. This means that you don't have to assign an additional library directory when you set up a new project to do OpenGL-related programming. Nothing else in the settings of the VS project need to change.

For those who do not have the *gl* directory, first try to find the *opengl32.dll* in the *system32* directory above. If you find it, then what you lack are only just the header files (*gl.h*, *glu.h*), and maybe *glu32.dll*. Try to find them on the internet and put them in the appropriate directories.

It's really rare that the *opengl32.dll* is missing, which is probably because your display card does not support OpenGL. Contact the TA if you have this problem, and we'll try to find a solution.

Although GLUT is a stable tool for creating OpenGL programs, its developers stopped to maintain it long time ago. As a substitute, FreeGLUT provides boths compatible and new functions. You can download from its official website (<http://freeglut.sourceforge.net/>) or from the Web-Learning. You may need to compile the code yourself, but the installation is the same as the original GLUT library.

3 Set up on Linux

There are many Linux distributions, and many of them contain OpenGL as a system component. We will describe the details of installation on Linux using Ubuntu as an example. If you're using another distribution, try compile the code at the end of the instruction directly to see if it works.

Type the following command in shell to install the basic function library:

```
$sudo apt-get install build-essential
```

Install the OpenGL library:

```
$sudo apt-get install libgl1-mesa-dev
```

Install the OpenGL utilities:

```
$sudo apt-get install libglu1-mesa-dev
```

Install the GLUT library:

```
$sudo apt-get install libglut-dev
```

At this step you might encounter a problem: you cannot find the package *libglut-dev* (probably because the original GLUT is kind of out-of-date). In this case, try to use the *freeglut3-dev* to replace the *libglut-dev* and run the command above again, it should be ok now.

After these steps, the environment you need should be set up. Try to compile the code at the end of the instruction with command like:

```
g++ test.cpp -o test -lGL -lGLU -lglut
```

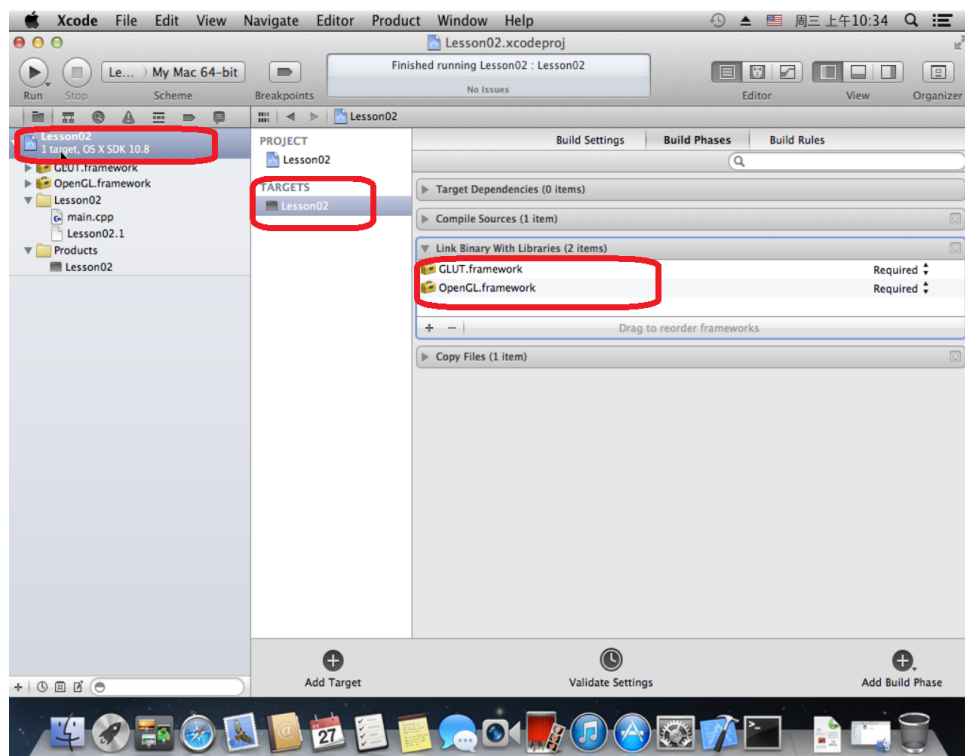
Run the generated bin to see if the result is right. Note that you need to link all the libraries we need in the compiling command (or in the makefile).

4 Set up on Mac

OpenGL is absolutely supported in Mac-OS, so if you're using Mac-OS, there is actually nothing you need to do here.

As in Windows, we suggest an IDE to do the course exercises and projects. XCode is a popular IDE under Mac-OS and you can choose it in this course.

What you need to do next is setting the properties of the project you're working on. Choose the project directory in the left column, and select the item below TARGETS in the middle. Under the "Build Phases" Tab on the rightside, there is a "Link Binary with Libraries" row. Add "OpenGL.framework" and "GLUT.framework". You can refer to the following figure for more details:



After this, you can try the program at the end of this instruction.

5 Test Code

Below is a tiny (maybe smallest) piece of code that creates a windows using GLUT library. Basically most of all exercises can be extended from it. Try it on your own computer to see whether it can be compiled and run. It will create a window with pure red background.

```
/*
Test code
*/

#include <gl\glut.h>

void myDisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glutSwapBuffers();
}

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

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Hello World");
    glutDisplayFunc(myDisplay);
    initFunc();

    glutMainLoop();

    return 0;
}
```