VG101 — Intoduction to Computers & Programming

Lab 7

Instructor: Manuel Charlemagne

Author: Yihao Liu

UM-SJTU JI – Summer 2019

Goals of the lab

- Install OpenGL

1 Introduction

Frank and Krystor are addicted to computer games, they found that most early games were developed by OpenGL. They wanted to try it, and are searching you for help installing it.

2 Working Flow

2.1 Install OpenGL

2.1.1 Windows (with MinGW)

Download FreeGlut For MinGW and you can get the following directory structure:

```
freeglut
   bin
       freeglut.dll
       x64
           freeglut.dll
   Copying.txt
   include
       GL
           freeglut_ext.h
           freeglut.h
           freeglut_std.h
           glut.h
   lib
       libfreeglut.a
       libfreeglut_static.a
       x64
           libfreeglut.a
           libfreeglut_static.a
   Readme.txt

6 directories, 12 files
```

Then you should copy some of these files into the corresponding MinGW directory (typically like this).

```
mingw64
   bin
   etc
   include
   lib
   libexec
   share
   ssl
   x86_64-w64-mingw32

8 directories, 0 files
```

Note

Make sure your mingw is 64-bit(mingw64) or 32-bit (mingw32) since compilers of different bits have different binaries.

The procedure for `mingw64` is:

i) copy the directory `freeglut/include/GL` into `mingw64/include`

ii) copy the file `freeglut/bin/x64/freeglut.dll` into `mingw64/bin`

iii) copy the files `freeglut/lib/x64/libfreeglut.a` and `freeglut/lib/x64/libfreeglut_static.a` into `mingw64/lib`

The procedure for `mingw32` is:

i) copy the directory `freeglut/include/GL` into `mingw32/include`

ii) copy the file `freeglut/bin/freeglut.dll` into `mingw32/bin`

iii) copy the files `freeglut/lib/libfreeglut.a` and `freeglut/lib/libfreeglut_static.a` into `mingw32/lib`

2.1.2 Windows (with MSVC)

Note

MSVC (Microsoft Visual C++) is not recommended in this course, so we don't have support on it. You can download FreeGlut For MSVC and follow the guide, but we won't help you solve any problem you meet.

2.1.3 Linux (Debian/Ubuntu)

Run the following commands and then OpenGL will be installed automatically.

```
$ sudo apt-get install build-essential
$ sudo apt-get install libgl1-mesa-dev libglu1-mesa-dev freeglut3-dev
```

2.1.4 MacOS (with LLVM/Clang)

Building FreeGLut is complex on MacOS and GLUT is originally contained in MacOS. Hence, we choose to make our life easier and use GLUT directly on Mac. There's nothing you need to do for environment setup.

2.2 Setup Build Environment

2.2.1 Build with CMake

Copy and paste the `CMakeLists.txt` provided by us.

```
cmake_minimum_required(VERSION 2.7)
project(OpenGL)

set(CMAKE_CXX_STANDARD 14)
set(CMAKE_CXX_FLAGS "-Wall -Werror -pedantic -Wno-unused-result
 ↪  -Wno-deprecated-declarations")

find_package(OpenGL REQUIRED)
include_directories(${OPENGL_INCLUDE_DIR})

find_package(GLUT REQUIRED)
include_directories(${GLUT_INCLUDE_DIR})

add_executable(OpenGL main.cpp)
target_link_libraries(OpenGL ${GLUT_LIBRARY} ${OPENGL_LIBRARY} m)
```

2.2.2 Build with XCode

i) Create a new CPP project with Command Line Tool.

ii) In the “Linked Frameworks and Libraries” area click the “+” button, and select “OpenGL.framework”

iii) Do the same thing for “GLUT.framework”

2.2.3 Build with Code::Blocks

Open Code::Blocks, find Settings -> Compilers -> Global Compiler Settings -> Linker Settings, hit Add at the bottom, change the folder to `mingw/lib`, add library files `libopengl32.a, libglu32.a, libfreeglut.a`, and hit Save.

2.2.4 Build with Command Line

Windows

```
g++ -Wall -Werror -pedantic -Wno-unused-result -Wno-deprecated-declarations
 ↪  -std=c++14 -o OpenGL main.cpp -lglu32 -lfreeglut -lopengl32
```

Linux

```
g++ -Wall -Werror -pedantic -Wno-unused-result -Wno-deprecated-declarations
 ↪  -std=c++14 -o OpenGL main.cpp -lglut -lGL -lGLU
```

MacOS

```
g++ -Wall -Werror -pedantic -Wno-unused-result -Wno-deprecated-declarations
 ↪  -std=c++14 -o OpenGL main.cpp -framework OpenGL -framework GLUT
```

2.3 Compile a Teapot Program

Here is a sample program of printing a Teapot with OpenGL.

```c
#ifdef __APPLE__
#include <GLUT/glut.h>
#elif _WIN32
#include <windows.h>
#include <GL/freeglut.h>
#else
#include <GL/freeglut.h>
#endif

void display() {

    /* clear window */
    glClear(GL_COLOR_BUFFER_BIT);

    /* draw scene */
    glutWireTeapot(.5);

    /* flush drawing routines to the window */
    glFlush();

}

int main(int argc, char *argv[]) {

    /* initialize GLUT, using any commandline parameters passed to the
       program */
    glutInit(&argc, argv);

    /* setup the size, position, and display mode for new windows */
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutInitDisplayMode(GLUT_RGB);

    /* create and set up a window */
    glutCreateWindow("hello, teapot!");
    glutDisplayFunc(display);

    /* tell GLUT to wait for events */
    glutMainLoop();
}
```

If you're able to compile and run it, OpenGL is set up successfully.

Something More

If you have carefully read through this documentation, you should have found that your code of OpenGL may not be portable (platform independent). Hence, for ease of our grading, please do clearly state in README on which plaform, Windows, MacOS or Linux you successfully compiled and ran your program.

3 Appendix

A guide from the last semester is also included. If you still have problems after reading through the lab, take the guide on the next few pages as a reference.

GLUT Setup Guldeline

Xiaohan Fu, Yue YAO, Zhengyang Feng, Chenggang Wu

2017/11/27

Contents

`Glut` stands for `Open[GL] [U]tility [T]ools`. It's a library designed to ease programmers work while dealing with OpenGL programmes. `OpenGL` stands for `Open Graphics Library`. As its name suggests, it is a library for drawing figures (and more) on the screen. Applications that use this technology includes, games (counter-strike), design softwares (AutoCAD, etc.), visual designing (Photoshop, Premiere, etc.), phones (Android smartphones), and Matlab!

This guildline aims to help Windows and MacOS users set up GLUT environment. For Linux users, we believe you could handle it by yourself.

1 Windows

GLUT hasn't been updated since 1998. That means the latest version of it might be even older than most of you. Hence, we choose its predecessor FreeGlut instead.

- Download FreeGLut for MinGW here. (The readme it offered may be to complex for you, just follow our instructions here)

You will see `/bin`, `/lib` and `/include` three directories. In these directories, there should be also a sub-directory `x64`, where files for 64-bit are stored. Outside is the 32-bit version files. Which one you should choose depends on the version of your gcc. Type in `gcc -v` in your terminal/cmd to check. For 32-bit gcc, use all 32-bit files in the following steps and vice versa.

- Copy the dynamic library file inside `/bin` (*.dll) into *both* `C:\Windows\System32` and `C:\Windows\SysWOW64` directory. *(same file in two directories)*
- Copy all header files (*.h) into your compiler's `include\GL` directory. In most cases, it should be something like `C:\MinGW\include\GL` or `C:\TDM-GCC\include\GL`.
- Copy all libraries (*.a) into compiler's `lib` directory. That is something like `C:\MinGW\lib\` or `C:\TDM-GCC\lib\`.

Strictly follow the above steps and you will be safe.

The next issue is compilation. To test, you could refer to the sample code in APPENDIX.

1.1 Compilation in Command Line

Assume there's a glut plotting code `plot.cpp`. To compile it, type the following in command line and will generate an executable file `plot`.

```
g++ -std=c++11 -pedantic -Wall -o plot plot.cpp -lglu32 -lfreeglut -lopengl32 -Wl,--subsystem,windows
```

Some thing need to notice: * `-lglu32` supports you using basic GLUT functions, which is mandatory in this course. * `-Wl,--subsystem,windows` ensures the executable runs as a Windows GUI application rather than a console application. * The linker flags `-lglu32 -lfreeglut -lopengl32` should always be in the tail of your compiling command to prevent some bug.

1.2 Configuration for CLion

For your project using FreeGlut, apply the following changes to your corresponding `CMakeLists.txt`.

Add one line after `add_executable` line, so that now it is

```
add_executable(project_name ...)
target_link_libraries(project_name opengl32 glu32 freeglut)
```

Make sure your `project_name` are consistent in both lines.

1.3 Configuration for Code::Blocks

Open Code::Blocks, find Settings -> Compilers -> Global Compiler Settings -> Linker Settings, hit Add at the bottom, change the folder to `...\MinGW\lib`, add library files libopengl32.a, libglu32.a, libfreeglut.a, and hit Save.

1.4 Including Libraries

Generally, you just need to follow the lecture slides using

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

But if you are an advanced user of FreeGlut and would use some extended functions offered by FreeGlut, we warmly welcome it and in that case, you should use

```
#include <GL/freeglut.h>
```

instead.

But under most conditions, your program would not be successfully compiled just with this. You need to try adding

```
#include <windows.h>
```

at the **very beginning** of your code. *Note that this order does matter.*

If you are tired of adding this line for all your source codes, one easier way is to add this line in `freeglut.h` or `glut.h` directly, which is also the recommended solution.

2 MacOS

Building FreeGLut is complex on MacOS and GLUT is originally contained in MacOS. Hence, we choose to make our life easier and use GLUT directly on Mac. There's nothing you need to do for environment setup. Then we talk about the compilation.

2.1 Compilation in Command Line

Things are a little bit different here. Assume there's a glut plotting code `plot.cpp`. To compile it, type in the following in command line and will generate an executable file `plot`.

```
g++ -std=c++11 -pedantic -Wall -o plot plot.cpp -framework OpenGL -framework GLUT
```

2.2 Configuration for CLion

Add the following line to your `CMakeLists.txt` and reload it.

```
set(CMAKE_EXE_LINKER_FLAGS "-framework OpenGL -framework GLUT")
```

2.3 Configuration for XCode

- Create a new CPP project with Command Line Tool.
- In the "Linked Frameworks and Libraries" area click the "+" button, and select "OpenGL.framework"
- Do the same thing for "GLUT.framework"

2.4 Including Libraries

Different from Windows, you should use

```
#include <GLUT/glut.h>
```

instead.

3 Something More

If you have carefully read through this documentation, you should have found that your code of OpenGL may not be portable (platform independent). Hence, for ease of our grading, please do clearly state in README on which plaform, Windows, MacOS or Linux you successfully compiled and ran your program.

4 APPENDIX

Sample code fetch from http://math.hws.edu/bridgeman/courses/324/s06/doc/opengl.html

```
// For MacOS
// #include <GLUT/glut.h>

// For Windows
// #include <windows.h>
// #include <GL/freeglut.h>

// Omit the warnings that some functions are out of date.

void display () {

    /* clear window */
    glClear(GL_COLOR_BUFFER_BIT);

    /* draw scene */
    glutSolidTeapot(.5);

    /* flush drawing routines to the window */
    glFlush();

}

int main ( int argc, char * argv[] ) {

    /* initialize GLUT, using any commandline parameters passed to the
       program */
    glutInit(&argc,argv);

    /* setup the size, position, and display mode for new windows */
    glutInitWindowSize(500,500);
    glutInitWindowPosition(0,0);
    glutInitDisplayMode(GLUT_RGB);

    /* create and set up a window */
    glutCreateWindow("hello, teapot!");
    glutDisplayFunc(display);

    /* tell GLUT to wait for events */
    glutMainLoop();
}
```