Student ID: 101062319
Name: 巫承威

# Homework 1 - Report

## Draw Some 3D Models

## • How to operate my program?

As the Figure 1. shown, there are several new header files and source files in my project, please make sure they are all under the project folder.

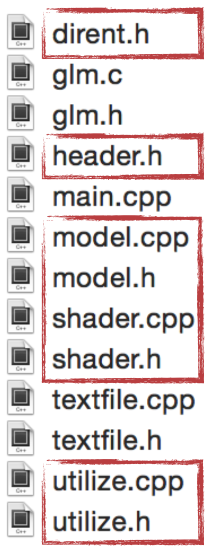
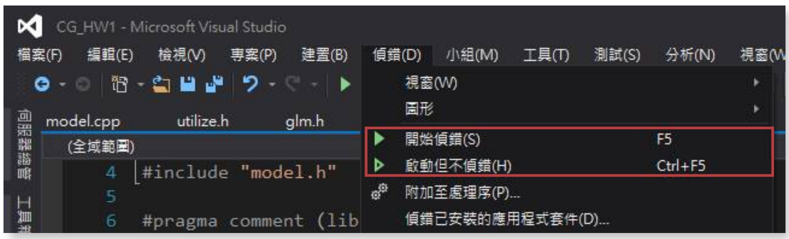

dirent.h
glm.c
glm.h
header.h
main.cpp
model.cpp
model.h
shader.cpp
shader.h
textfile.cpp
textfile.h
utilize.cpp
utilize.h

**Figure 1.**
Header Files and Source Files


CG_HW1 - Microsoft Visual Studio
檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 工具(T) 測試(S) 分析(N) 視窗(W)
視窗(W)
圖形
開始偵錯(S) F5
啟動但不偵錯(H) Ctrl+F5
附加至處理序(P)...
偵錯已安裝的應用程式套件(D)...
model.cpp
utilize.h
glm.h
(全域範圍)

```
#include "model.h"

#pragma comment (lib
```

**Figure 2.**
ScreenShot of Microsoft Visual Studio 2013

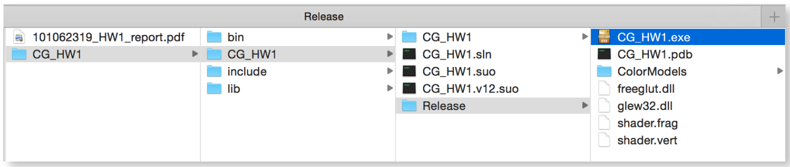


**Figure 3.**
Path of the "CG_HW1.exe"

All the user need to do to run my program is just open the .sln poject file via Microsoft Visual Studio 2013 (higher version may be fine). Then click the buttons "開始偵錯" or "啟動但不偵錯", and the execution window will appear. Or just execute the "CG_HW1.exe" as the Figure 3. shown. Like the example that TAs provided, user can type some button to control the display window and the object on it. These are the commands:

- h - help menu or information
- c - clear console
- w - switch between solid / wired
- z/x - previous / next model

## • How do I normalize models to [-1, 1]?

First, I traverse all the vertices of the object read from the function "glmReadOBJ" to get some statistics:

- Max values of three dimensions (Max X, Max Y, Max Z)
- Minimum values of three dimensions (Min X, Min Y, Min Z)
- Max length of three dimensions (Max X-length, Max Y-length, Max Y-length)

After getting these values, I can get the central point of the whole model, and I shift the whole model via traversing all vertices and minus the displacement between the model's central point and the original point. Then, I shift the model to the central position of the window. Next, I pick the max length among all the displacement of three dimensions to be divided by 2 which is the range from -1 to 1 to get the value of the scale. Same with the above process, I traverse all the vertices and divide themselves by the scale. And to make the above two processes run efficiently, I combine the shift process with the scaling process. Finally, I get the correct points of the whole model.

## • Implementation and problems I met

At the beginning, I draw the models via running a for loop to draw the every triangle of the model which means I traverse the model's triangle array to get each triangle's information and to draw it via calling the function "glDrawArrays" in each round of loop. However, this method seems to be an inefficient way since it need to call the function "glDrawArrays" in each round of loop. Due to this concern, I modify the drawing process which will allocate two big float arrays first with the size of the number of the model's triangles times nine. Next, I traverse all the triangles of the model to fill the correct order of all the vertices which will be sent to the drawing function "glVertexAttribPointer" sooner. And it can fix the problem of calling "glDrawArrays" too many times since it only need to call it once. Another problem I met is how to add the keyboard features. At first, I come up with a method that I just only need to store them in a big character array of each file's name and this method is somehow called brute-force method. Fortunately, I saw the discussion on the iLMS's forum which is about the usage of the structure "DIR" and "dirent". I decide to learn how to use it, so I change the previous method. "DIR" and "dirent" are the structure used in Linux system, so if I want to use them in my homework, I need to include the library called "dirent.h". Also, this way can not only get all the models' file name but also can traverse from the given folder name to its subfolders and get all the files' name. It's more powerful and flexible than the brute-force method. Thanks for the discussion on the iLMS. :)

## • Other efforts I have done

Owing to the unfriendly codes which combine all the segment of the program into only one file, and also not so readable for the programmer, I decide to make it looked like the object-oriented program. Nevertheless, this idea cost me almost one afternoon to adjust them…. To finish this task, I need to figure out the operation between the model and the whole program and also how to manipulate each class. I think this is the most difficult part for transforming the original code to object-oriented type.

Another part I have done is to use the structure “DIR” and “dirent” which are already mentioned in above paragraph. It’s more powerful and flexible than brute-force method because it can traverse all the files in the given folder’s all subfolders. Also, I define new macro called “abs” to help get the length of the model. To avoid the re-define error of so many new header files, I add the “#pragma once” to make compiler correctly include the header file only once when they need.

What’s more, I add the feature that my model can be scaled by adjusting the GLUT window. It means that if I change the GL window, my object model will also change its size to fit the size of window just like Figure 5. shown.

## • Screenshots

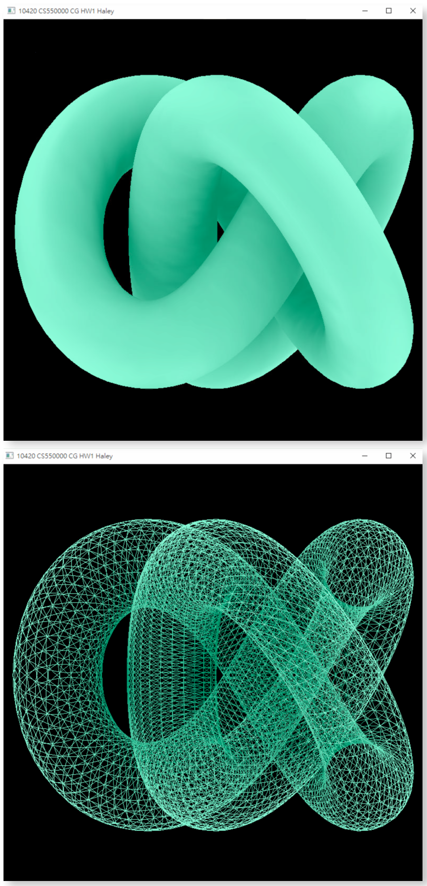
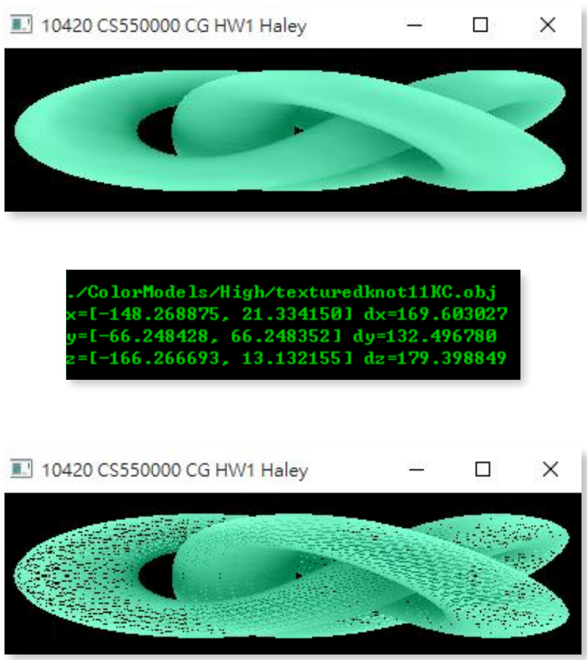


**Figure 4.**

Fill Mode & Line Mode of the Model

“Texturedknot11KC.obj”



```
./ColorModels/High/texturedknot11KC.obj
x=[-148.268875, 21.334150] dx=169.603027
y=[-66.248428, 66.248352] dy=132.496780
z=[-166.266693, 13.132155] dz=179.398849
```

**Figure 5.**

Different Scale of the Model

“Texturedknot11KC.obj”