



Qatar University

College of Engineering

Department of Computer Science and Engineering

Computer Graphics HW#4 (Fall 2021) Self-Learning Work

Course

Name, ID, Section: Computer Graphics CMPS 373

Instructor Name: Dr. Osama Halabi

HW Info

Due: Nov. 10, 2021 (Wed)
(End of the day)

Grade: 100 marks

100

Student

Student Name:

Student ID:

SUBMISSION INSTRUCTION

You must submit the whole homework solution/project folder in one zip file.

Name your file as [your name]-[student ID]-HW3

Before creating the zip file for your solution go to Build -> Clean Solution to remove linking files.

All programming projects should follow the same file organization used in the lectures. Therefore, all the following items are expected to be included:

1. all C++ source (.cpp) files (in \src)
2. all C++ executable (.exe) files (in \bin)
3. all shader source (GLSL) files (in \shaders)
4. all necessary supporting files such as models, textures, cube maps, normal/height maps
5. a "readme" file describing your program, including the items listed below:
 - a. a screen capture of your running program.
 - b. a list of required items that you were not able to complete.
 - c. citation and copyright permission information for all elements used by your program.
6. you **must set the title of the window to the homework number and your name. For example, "CG HW1 Osama Halabi"**.
7. This is an **individual work** and the submitted work must represent your **own thinking and efforts**. Copying the work of others will not be tolerated. Similarity will be investigated and zero grade for all parties involved will be assigned.
8. Submit before the due date.

Description

You have learned how to load 3D model from OBJ format. The wavefront object format is generally considered to be an easy-to-parse model format. However, the loader was simple and do not support reading only one UV unwrapped texture file. Also, do not support reading materials as well.

A very popular model importing library out there is called Assimp that stands for *Open Asset Import Library*. Assimp is able to import dozens of different model file formats (and export to some as well) by loading all the model's data into Assimp's generalized data structures. As soon as Assimp has loaded the model, we can retrieve all the data we need from Assimp's data structures. Because the data structure of Assimp stays the same, regardless of the type of file format we imported, it abstracts us from all the different file formats out there.

Your task is to implement the model loading using Assimp library.

1. Create Mesh class (Mesh.h)

Set the minimal requirements for a mesh class we can define a vertex in OpenGL.

2. Create Model class (Model.h)

The goal of this chapter is to create another class that represents a model in its entirety, that is, a model that contains multiple meshes, possibly with multiple textures.

3. Create mainModelLoading

Demonstrate the use of the loader by implementing model loading.

Hint:

You can utilize the following site to learn how to implement this task. The site describes in details all the required steps and even provides the code, however, you have to understand that we are implementing in different coding style using classes, so you can easily learn how to do it and just use the methods that we implemented that related to shaders and textures loading.

<https://learnopengl.com/Model-Loading/Assimp>