Computer Graphics Class Assignment 1

2015004493

김형순

1. How to run program

a. You can open OBJ file with drag – and – drop to viewer window.

a-1. I found the OBJ file format has various type in representing face information. But in this assignment, I handled file with vertex normal information in face information. So, if you drop file with no vertex normal information, you can’t use [shading using normal data in OBJ file] mode.

b. You can change wireframe / solid mode by pressing Z key.

c. You can also change shading mode of shading using normal data in file and forced smooth shading by pressing S key. I changed color of light when pressing S to distinguish shading mode. (shading using normal data in file -> light blue, forced smooth shading -> light red)

d. You can also see the number of vertexes and faces and file name on console window.

e. The default mode of viewer is solid mode (can change by Z key) and forced smooth shading mode (can change by S key).

2. Implemented requirements

a. Manipulate the camera

I used my assignment 1 code. You can rotate, move, zoom camera with mouse buttons and wheel.

b. Load an OBJ file and render it

i) I made viewer with drag and drop properties. When you drop file, it may take some time to render object, but you can see result after a while.

ii - 1) I parsed data in OBJ file, and stored it to global variables, which is vertex\_array, normal\_array, index\_array, and face\_normal\_array. You can see this code part in drop\_callback function.

ii – 2) The stored data will be handled in create\_vertex\_seperate function and calculate\_normal function. The previous one is function for making vertex array with each vertex’s normal. The last one is function for making vertex array, index array, and calculate each vertex’s normal and store to normal array. When calculations and making arrays are done, the viewer will render object on your window. I used glDrawArrays when using normal data in file and used glDrawElements when using my own calculated normal. (forced smooth shading).

ii -3) I only used three vertexes, and when polygon has more than 3 vertexes, I just didn’t use remains.

iii) I made property of toggling wireframe / solid mode by pressing Z key. When Z key is pressed, polygon\_mode variable will be changed and in render function, this variable is used deciding whether toggle wireframe mode or solid mode.

iv) You can see following data on console: File name, Total number of faces, Number of faces with 3 vertices, Number of faces with 4 vertices, Number of faces with more than 4 vertices.

v) I also made property of toggling [shading using normal data in OBJ file] / [forced smooth shading] by pressing S key. Previous one is rendered in draw\_vertex\_seperate() function, and last one is rendered in draw\_forced\_smooth\_shading() function.

C. Lighting

I made lighting() function to make multiple lights. I used 5 different lights in this view, lights’ color is changed with shading mode. [shading using normal data in OBJ file] mode is light blue, [forced smooth shading] is light red.

I used five lights in position with following positions.

(0., 6., 3., 1.)

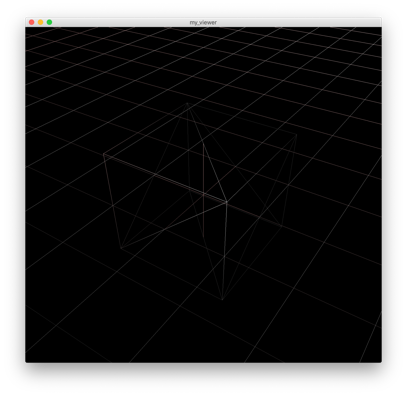
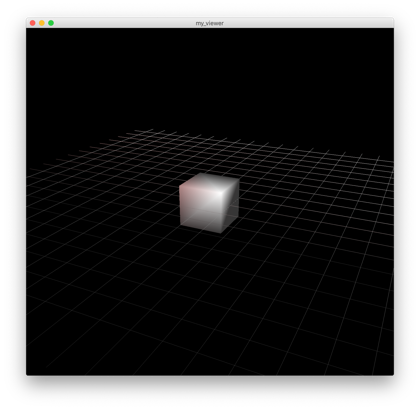
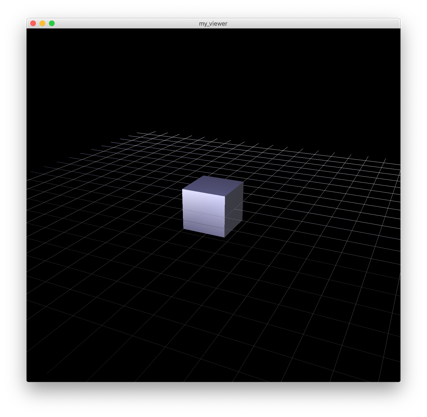
(-1.5, 3., 3., 1.) (1.5, 3., 3., 1.)

(-3., 1.5, 3., 1) (3., 1.5, 3., 1.)

With multiple trials I think those position can show the object and shadow properly, appropriately. I used all lights with point light, because in my tests, point light is more efficient than directional light when represent shading.

D. A few screenshot images

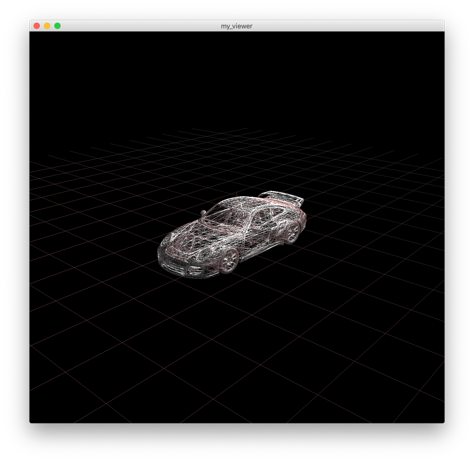
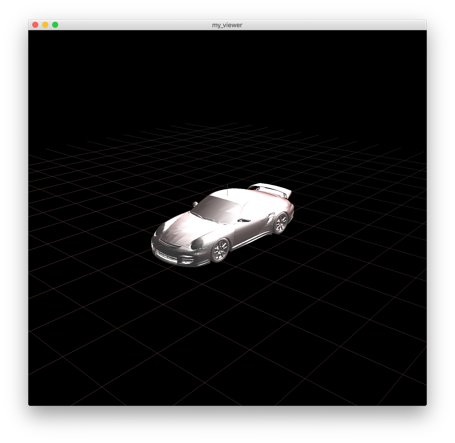
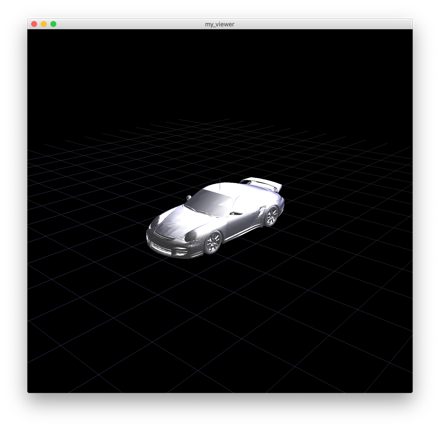
Cube



shading using normal forced smooth shading wireframe mode

data in OBJ file

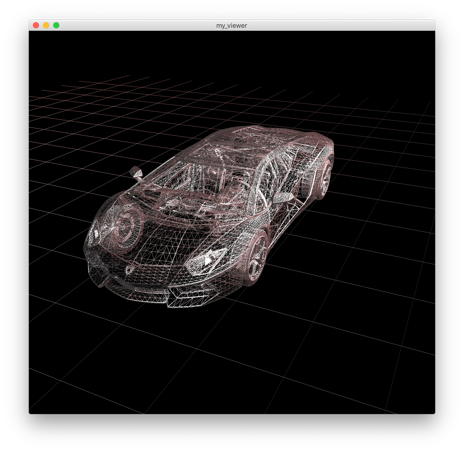
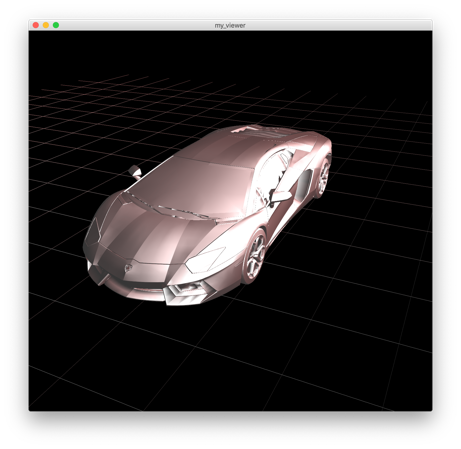
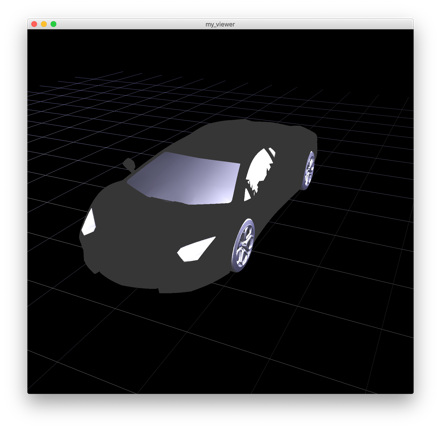
Porsche : If polygon is dense enough, there are few differences between shading using normal data and forced smooth shading.



shading using normal forced smooth shading wireframe mode

data in OBJ file

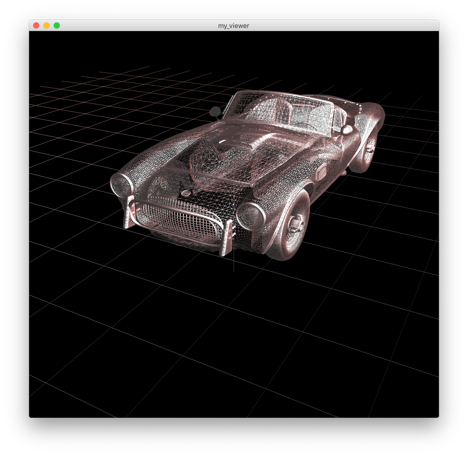
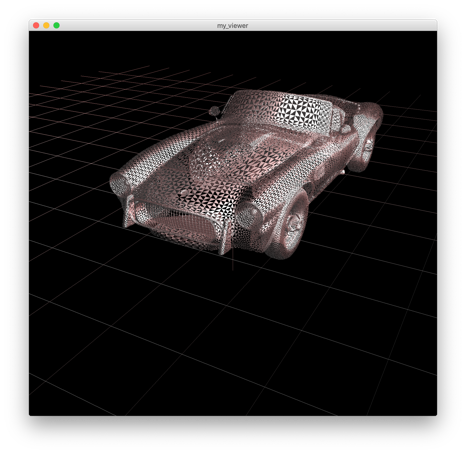
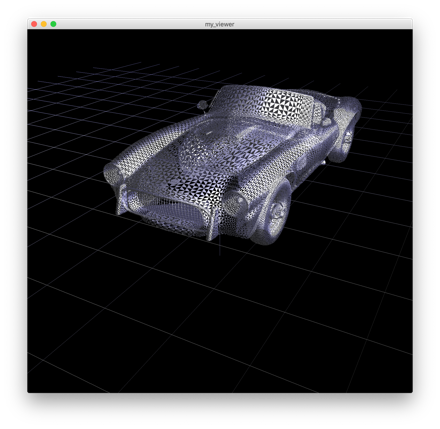
Avent : There was insufficient normal data in obj file. But by forced smooth shading, model was rendered successfully.



shading using normal forced smooth shading wireframe mode

data in OBJ file

Shelvy : If polygon has more than 4 vertices, program just render with 3 vertices (Triangle)



shading using normal forced smooth shading wireframe mode

data in OBJ file