Computer Graphics Assignment 3: OBJ Viewer

Handed out: May 24, 2018

Due date: 23:59, June 13, 2018 (NO SCORE for late submissions!)

Submit your assignment only through the Assignment (과제) menu of the lecture home at portal.hanyang.ac.kr.

1. Implement an obj file viewer for triangle meshes. The loaded mesh should be rendered with multiple light sources.

2. Requirements

A. Manipulate the camera in the same way as in Lecture 17, but add zoom in / out feature (10 pts)

i. Rotation & Translation: Just reuse the code in 17-code.py

1. Key 1 & 3: rotate the camera about y axis
2. Key 2 & W: translate the camera along the y axis

ii. Zoom: Move the camera forward (close to the origin (0,0,0): zoom in) or backward (far from the origin: zoom out)

1. **Key A: zoom in**
2. **Key S: zoom out**

iii. Use perspective projection

B. Load an obj file and render it (70 pts)

i. Open an obj file by drag-and-drop to your obj viewer window (10 pts)

1. Google *glfwSetDropCallback* to see how to do it
2. The viewer should only render one obj file at a time. If an obj file B is drag-and-dropped to the viewer while it is rendering another obj file A, the viewer should only render the new obj file B.

ii. Read the OBJ file and display the mesh only using vertex positions, vertex normals, faces information (40 pts)

1. Ignore texture coordinate, material, group, shading information. In other words, ignore vt, mtllib, usemtl, o, s tags
2. You can use either a set of glVertex*() & glNormal*() calls or a vertex array (see Extra Credits section) to render the mesh

iii. Toggle wireframe / solid mode by pressing **Z key** (similar to pressing Z key in Blender) (10 pts)

iv. When open a new obj file, print out the following information of the obj file to stdout (console) (10 pts)

1. File name
2. Total number of faces
3. Number of faces with 3 vertices
4. Number of faces with 4 vertices
5. Number of faces with more than 4 vertices

C. Lighting (10 pts)

i. Use multiple light sources (not a single light) to better visualize the mesh (10 pts)

ii. Choose the number of light sources, light source types, light colors, material colors as you want

3. Report (10 pts)

A. Submit a report of at most 3 pages in docx file format (MS Word). Do not exceed the limit.

B. The report should include:

i. A few screenshot images (2~3 images)

1. Download cool obj files from the Internet and open them with your viewer. You may download obj files from

A. https://free3d.com/

B. https://www.cgtrader.com/free-3d-models

2. Include some of the best looking screenshot images of your viewer when the downloaded obj files are opened

ii. Explanation about:

1. How to run your program

2. Which requirements you implemented

3. Lighting configuration: how many light sources? where do you put the light sources? what is the type of each light source?

4. Extra credits (Only one of three items A, B, C will be reflected in the score)

A. Load & render a mesh that does not have the same number of vertices of all polygons **(+10 pts)**

i. For example, some polygons in the mesh are triangles and the rest are quads or polygons with more vertices

B. Use glDrawArrays() or glDrawElements() to render a triangle mesh **(+10 pts)**

C. Load & render a mesh that does not have the same number of vertices of all polygons using glDrawArrays() or glDrawElements() **(+20 pts)**

i. To do this, you have to render a quad or a n-polygon as a set of triangles. So you may need some kind of "triangulation" algorithm.

5. Your program should be able to run on systems with Python 3.5, NumPy, PyOpenGL, PyOpenGL_accelerate, glfw. Do not use any other additional python modules.

6. Test your viewer with sample obj files

A. cube-tri.obj: A cube with triangles only

B. cube-tri-quad.obj: A cube with triangles and quads

C. sphere-tri.obj: A sphere with triangles only

D. sphere-tri-quad.obj: A sphere with triangles and quads

E. cylinder-tri.obj: A cylinder with triangles only

F. cylinder-tri-quad-n.obj: A cylinder with triangles, quads and polygons with more vertices

G. Basically, your viewer should be able to render cube-tri.obj, sphere-tri.obj, cylinder-tri.obj properly

H. To meet extra credit requirement A and C, your viewer should be able to render all above sample obj files properly

7. What you have to submit:

A. **A zip file** including

i. **.py files**

1. You can use multiple .py files for this assignment. In this case, explain how to run the program in the report.

ii. **.docx report file**

8. Additional information

A. *drop_callback* in glfw python binding is slightly different from that of original glfw written in C. *drop_callback* in python takes only two parameters, *window* and *paths*. *paths* is a list of dropped file paths.

B. OBJ file format reference: https://en.wikipedia.org/wiki/Wavefront_.obj_file

C. Python provides powerful string methods helpful for parsing an obj file. Among them, split() will be most useful.