Computer Graphics Project 2: Obj viewer & drawing a ierarchical model

Handed out: April 17, 2023

Due: 23:59, May 14, 2023 (NO SCORE for late submissions!)

- *LMS course home > Lecture Contents (강의콘텐츠) > Week 5> Project 1 > "Submit Assignment" Button> Upload your zip file*
- *Compress your files into a zip file as in the following example. The zip file can be named whatever you want.*

```
+ submission.zip
  - main.py
  - ...
  - report.pdf
```

- *Your program may consist of several python source files. But the main module should be in **main.py**. That is, your program should be executed with the following command:*

```
python main.py
```

1. Implement your own obj file viewer 1) showing a single loaded obj mesh and 2) showing an animation of a hierarchical model consisting of loaded obj meshes. The multiple light sources should be used for rendering.
   A. You must implement all requirements in a single program. This project DOES NOT require each requirement to be a separate program.
   B. Your program should run in two modes – **"single mesh rendering mode"** and **"animating hierarchical model rendering mode".**
   C. The window size doesn't need to be (800, 800). Use the larger window that is enough to see the details of the viewer.
   D. **Your program must use OpenGL 3.3 Core Profile**, meaning that …
      i. Your python code must include:

```
glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 3)
glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 3)
glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE)
```

ii. Your shader code must start with:

```
#version 330 core
```

iii. **You will not get any score for this project (except report) if you do not use OpenGL 3.3 Core Profile.**

E. Total points: 150 pts

2. Requirements

A. Manipulate the camera in the same way as in Project1 using your Project1 code (10 pts).

i. Also draw the reference grid plane.

B. Single mesh rendering mode (50 pts)

i. When a user does a drag-and-drop action on your viewer, your program should run in **"single mesh rendering mode"**.

ii. Open an obj file by drag-and-drop to your obj viewer window.

1. Google *glfwSetDropCallback* to see how to do it.
2. The viewer should render only 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.
3. **This feature is essential for scoring your assignment, so if not implemented, you won't get any score for "Single mesh rendering mode (50 pts)".**

iii. 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.

iv. When open an obj file, print out the following information of the obj file to stdout (terminal) **(10 pts)**

1. Obj 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. **Animating hierarchical model rendering mode (50 pts)**

i. When a user **presses a key 'h'** on your viewer, your program should run in **"animating hierarchical model rendering mode"**.

1. If a user drag-and-drops an obj file in this mode, your program should re-enter "single mesh rendering mode".

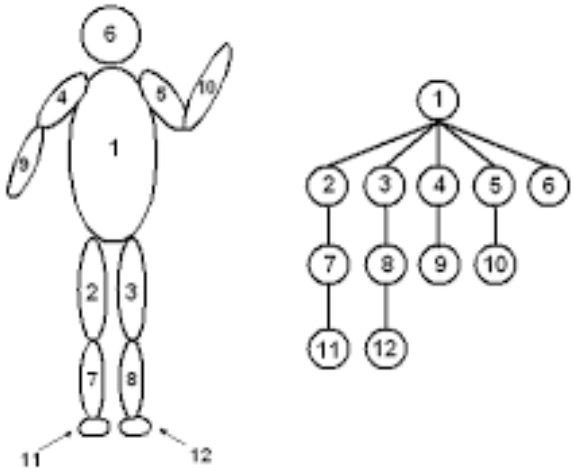
ii. The model should consist of **at least 3 different meshes loaded from 3 different downloaded obj files (10 pts)**.

1. **You MUST include the obj files used for this requirement in your submission zip file and use "relative paths" to specify those files in your source code.**
   A. Test your final submission files by putting them in a directory different from your working directory and run the program, before submitting them.
   B. The "relative paths" in the source code should not be platform-dependent. **Use os.path.join() for platform-independent path joining.**
   C. **If this part of the program does not run normally for any reason, you won't get any score for "Animating hierarchical model rendering mode (50 pts)".**
2. Download cool free obj files from the Internet and use them. For example,
   A. https://free3d.com/
   B. https://www.cgtrader.com/free-3d-models
3. **Do not use the provided sample obj files for this requirement.** You must use other obj files downloaded from internet to get score for this requirement.

Otherwise, **you won't get points for this requirement (that is, -10 pts).**

iii. The model should have a **hierarchy of 3 levels** and **each node** (except leaf nodes) **should have at least 2 child nodes**. **(15 pts)**.

1. For example, the following model has a hierarchy of 4 levels, and node 1 has five children but node 2--8 have only one child (so does not satisfy this requirement).

A. 

2. **All nodes should be visible. Otherwise, you'll get -10 pts.**

iv. **Animate the model** to show the hierarchical structure **(25 pts)**.

1. **ALL child body parts should move relative to their parent body part.**

A. In the above example, part 2, 3, 4, 5, 6 should move relative to part 1, part 7 should move relative to part 2, part 11 should move relative to part 7, … and so on.

B. If any of the child bodies does not move relative to its parent, you will get -15 pts.

2. The model should be **automatically animated without any mouse or keyboard inputs.**

3. You can make any hierarchical system freely. Be creative.

D. Lighting & Etc (25 pts)

i. Render all object using Phong Illumination and Phong shading. **(10 pts).**

1. Choose lighting parameters (light colors, light position, material colors, material shininess, …) as you want.

ii. Use multiple light sources **(10 pts)**

iii. Toggle wireframe / solid mode by pressing **'z' key** (similar to pressing 'z' key in Blender) **(5 pts).**

1. This feature should be available in both **"single mesh rendering mode"** and **"animating hierarchical model rendering mode"**.

3. Report (15 pts)

A. Submit a report of **at most 2 pages** in a **pdf** file. Using MS Word is recommended. Do not exceed the limit.

B. The report should include:

i. Which requirements you implemented **(5 pts)**

ii. A hyperlink to the video uploaded to Internet video streaming services (such as YouTube and Vimeo) by capturing the animating hierarchical model as a video **(10 pts).**

1. **The uploaded video MUST be publicly accessible.** Otherwise, you won't get the 10 pts.

C. You do not need to try to write a long report. Just only write down the required information. Use either English or Korean.

4. Runtime Environment

A. Your program should be able to run on Python 3.8 with only NumPy, PyOpenGL, glfw, PyGLM installed. Do not use any other additional python modules.

B. Only **glfw** is allowed for event processing and window & OpenGL context management. **Do not use glut functions for this purpose.**

C. If your program does not meet this requirement, it will not run on TA's computer so you will not get any score for this project (except report).

5. Description of the sample obj file for the "single mesh rendering mode" test

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

6. What you have to submit: A zip file including

A. **.py files -** *Your program may consist of several python source files. But the main module should be in* ***main.py****.*

B. **.obj files -** The obj files used for "Animating hierarchical model rendering mode".

C. **.pdf report file**

7. 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.