

## Computer Graphics Assignment 2: Obj viewer & drawing a hierarchical model

Handed out: April 20, 2022

**Due: 23:59, May 17, 2022 (NO SCORE for late submissions!)**

- Only files submitted by **git push to this course project** at <https://hconnect.hanyang.ac.kr> (<Year>\_<Course no.>\_<Class code>/<Year>\_<Course no.>\_<Student ID>.git) will be scored.
- Place your files under the directory structure **<Assignment name>/<your files>** just like the following example.

```
+ 2020_ITE0000_2019000001
+ ClassAssignment1/
- main.py
- report.docx
```

- The submission time is determined not when the commit is made **but when the git push is made**.
  - Your files must be committed to the **master branch**. Otherwise, it will not be scored.
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 have to implement all requirements in a single program. This assignment DOES NOT require each requirement to be a separate program.
    - B. The window size doesn't need to be (480, 480). Use the larger window that is enough to see the details of the viewer.
    - C. Your program should run in two modes – "single mesh rendering mode" and "animating hierarchical model rendering mode"
    - D. **DO NOT** set the window title to **your student ID**.
    - E. Total points: 155 pts (If you meet all the extra credit requirements, 20 pts are added)

## 2. Requirements

### A. Manipulate the camera in the same way as in ClassAssignment1 using your ClassAssignment1 code (10 pts).

- i. Also draw the reference grid plane.

### B. Single mesh rendering mode (50 pts)

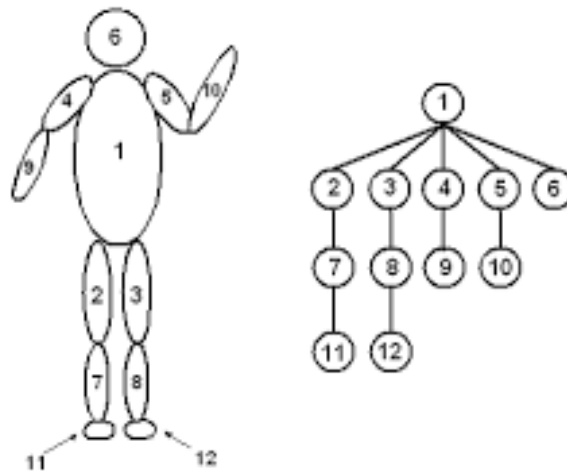
- i. When a user does a drag-and-drop action on your viewer, your program should run in **"single mesh rendering mode"**.
- ii. Open an obj file by drag-and-drop to your obj viewer window.
  1. Google *glfwSetDropCallback* to see how to do it.
  2. The viewer should render only one obj file at a time. 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.
  2. Use `glDrawArrays()` or `glDrawElements()` to render triangle meshes.
    - A. **DO NOT use `glVertex*()` & `glNormal*()`.** If you draw meshes using `glVertex*()` & `glNormal*()`, you'll get only **10 pts** out of 40 pts.
- iv. When open an 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. **Animating hierarchical model rendering mode (55 pts)**

- i. When a user **presses a key 'h'** on your viewer, your program should run in **"animating hierarchical model 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 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 public obj files from the Internet and use them. For example,
  - A. <https://free3d.com/>
  - B. <https://www.cgtrader.com/free-3d-models>
- iii. **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'll get -20 pts for this requirement.**
- iv. **You should use OpenGL matrix stack** to draw and animate your hierarchical model. **(10 pts)**.
- v. 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 2,3,4,5,6,7,8 have only one child (so does not satisfy this requirement).



A.

2. **All nodes should be visible. Otherwise, you'll get -5 pts.**

vi. **Animate the model** to show the hierarchical structure **(20 pts)**.

1. **ALL child body parts should move relative to their own 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 -10 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 (20 pts)

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

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

ii. Toggle wireframe / solid mode by pressing '**z**' key (similar to pressing 'z' key in Blender) **(10 pts)**.

1. `glPolygonMode( GL_FRONT_AND_BACK, GL_LINE )` # call this at the beginning of your render function to draw in wireframe mode.

### 3. Report (20 pts)

- A. Submit a report of **at most 2 pages** in docx file format (MS Word). 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.
  - iii. Lighting configuration **(5 pts)**:
    - A. How many light sources?
    - B. Where do you put the light sources?
    - C. What is the type of each light source (point light or directional light)?
- ⊖ You do not need to try to write a long report. Just only write down the required information. Use either English or Korean.

### 4. Extra credits

- A. Toggle [shading using normal data in obj file] / [forced smooth shading] by pressing '**s**' **key (+10 pts)**
  - i. In [forced smooth shading] mode, do not use vertex normal in obj. Instead, you have to compute the averaged vertex normal for each vertex as described in the lecture slide and use them for shading.
- B. Load & render a mesh that does not have the same number of vertices of all polygons using `glDrawArrays()` or `glDrawElements()` **(+10 pts)**
  - i. For example, some polygons in the mesh are triangles and the rest are quads or polygons with more vertices.
  - ii. To render this kind of mesh using a vertex array, you might need to render a quad or a n-polygon as a set of triangles. So you may need some kind of "triangulation"

algorithm.

## 5. Runtime Environment

- A. **Your program should be able to run on systems only with Python 3.7 or later, NumPy, PyOpenGL, glfw. 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 assignment (except report).**

## 6. Sample obj files the “Single mesh rendering mode”.

- 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 the extra credit, your viewer should be able to render all above sample obj files properly.**

## 7. What you have to submit:

- A. **.py files**
  - i. You can use multiple .py files for this assignment. In this case, explain how to run the program in the report.
- B. **.obj files**

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

C. **.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](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.