

CGRA354 Computer Graphics Programming T1/2024

Assignment #1: Geometry and colour (10 points)

Assigned: Tuesday, 27th February

Due: 14th March 2024 11:59pm

The areas addressed in this assignment include the use of Object Oriented Paradigms, C++ language and basic Shading in Computer Graphics programming.

Codebase

You are provided with a cross-platform programming framework specifically designed for the Computer Graphics assignments. The framework performs the initialization of the OpenGL pipeline allowing to focus on the assignment tasks. Comprehensive instructions for building the framework on different platforms can be found in the `README.md` file located in the source code root folder. The initial configuration of the project is displayed in `Initial_setup.png` file.

Turn in procedure

You should submit your work as a zip file using the ECS submission system. Please name your file as `<LastName><FirstName>-Assignment<X>.zip` where X is the assignment number. When your file is unzipped you should have:

1. The C++ code and programs you have written. You should use files in the form of the samples given, rather than producing files from scratch. This will help us follow your code.
2. Sample images created by your program. You can save these by taking a screenshot or use a screen capture tool.
3. A `readme.md`, `txt`, `pdf` or MS Word file that answers any questions posed, and that lists the input used to create the images you include. Alternatively you can provide a narrated screen-capture video showing the functionality of your program.
4. Any other information or supporting documentation to help us run and evaluate your submission.

If your programs fail on the machines used for grading, you may be asked to bring in your system to demonstrate that the files you submitted functioned in the environment you worked in.

Grading rubrics

There are multiple questions in this assignment, with some question having multiple sub-parts corresponding to a particular tier (core, completion, challenge and writing).

Part 1: Geometry and colour (10 points)

In this exercise, you will write a C++ class that implements a model loader for a specific 3D geometry definition file format (OBJ). This loader will be primarily a parser for Wavefront format files which are simple text files widely adopted in 3D graphics application [1, 2, 3]. You will be working with OpenGL commands for drawing with a basic GUI support in order to specify a name/path of an .obj file to load and the model colour to render with.

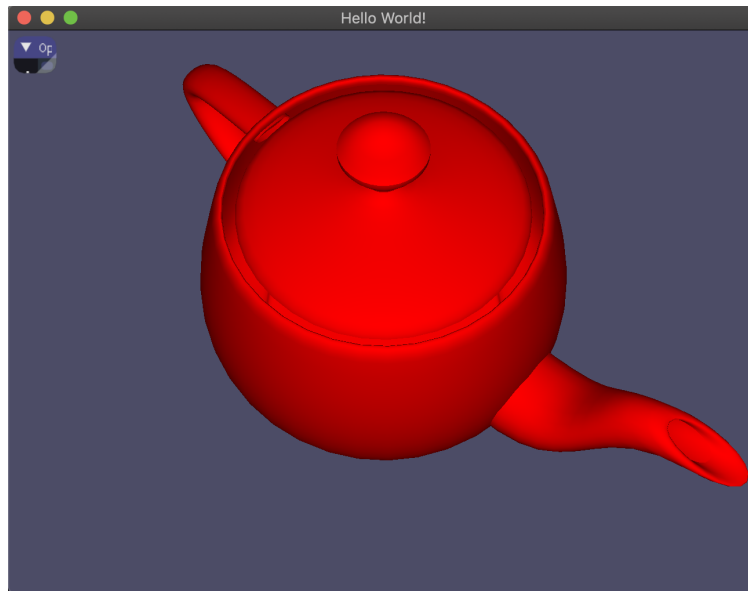


Figure 1: A rendering of the Utah teapot (1975, by Martin Newell) model by CGRA251 framework.

Implement a C++ class that parses and draws OBJ files according to the Wavefront format description. Name your class `ObjFile`, and implement it in files `objfile.h` and `objfile.cpp`. You are to implement the following class members:

Core (5 points)

- Private variables to store vertices (positions and normals) and the surface primitives (triangle indices).
- Public function, `loadOBJ()`, with a single string argument of the filepath to the .obj file, that loads the data from the file into the private variables. Let the file name be a string specified in the ImGui input controls.
- Public functions `build()`, `draw()`, and `destroy()` which set up, draw, and clear mesh geometry data buffers in OpenGL, respectively.
- Public function, `printMeshData()`, which prints out the (current) raw mesh data in the console. Link the `printMeshData()` function to a button in ImGui. (This will be useful for debugging).

Completion (3 points)

Add controls to the ImGui window to allow the user to interactively select the colour of the displayed model. This will involve:

- Add ImGui controls to allow the input of model colour (RGB values).
- Modifying the shader program, to accept selected colour values and apply them to the loaded model.

Challenge (2 point)

Improve the lighting controls to allow the user to interactively set the position/direction of the light source for displaying the model. This will involve:

- Adding ImGui controls to allow the input of either the light position or direction.
- Modifying the shader program, to accept the light settings and apply them to the shading for the loaded model.

Note: You must use the C++ I/O streaming and memory management facilities on this assignment.

References

- [1] Wiki: Wavefront .obj file,
https://en.wikipedia.org/wiki/Wavefront_.obj_file
- [2] CS Utah: Object Files (.obj),
http://www.cs.utah.edu/~boulos/cs3505/obj_spec.pdf
- [3] OBJ File Format – Simply Explained,
<https://all3dp.com/1/obj-file-format-3d-printing-cad/>