CGRA350 Real-time 3D Computer Graphics T2/2020

Assignment #1: Shader programming (20 points)
Assigned: Tuesday, 14th July
Due: 3rd August 2020 11:59pm

The areas addressed in this assignment include parametric generation of basic shapes, writing and understanding shaders, applications of textures and shadow mapping.

Codebase

The source code of this assignment should be familiar to you from the CGRA251 course. 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 `Assignment_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++ and shader 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. Your report in PDF format (either typed directly or scanned in).
3. If the question asks you to write code to make images, provide sample images created by your program. You can save these by taking a screenshot. Alternatively you can include these in your PDF report.
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

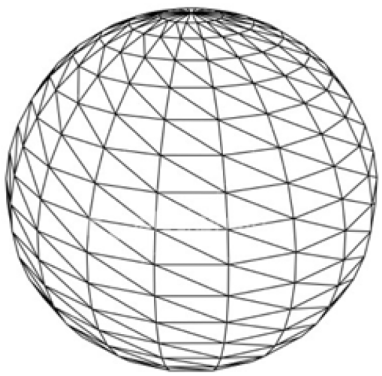
There are multiple questions in this assignment, with each question having multiple sub-parts corresponding to a particular tier (core, completion, challenge and writing).

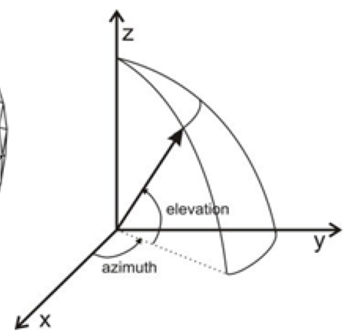
Part 1: Geometry (7 points total)

Representing simple geometric primitives. Implement two routines to approximate a sphere primitive, `sphere_latlong` and `sphere_from_cube`, and a routine to generate a torus primitive, `torus_latlong`.

Core (4 points)

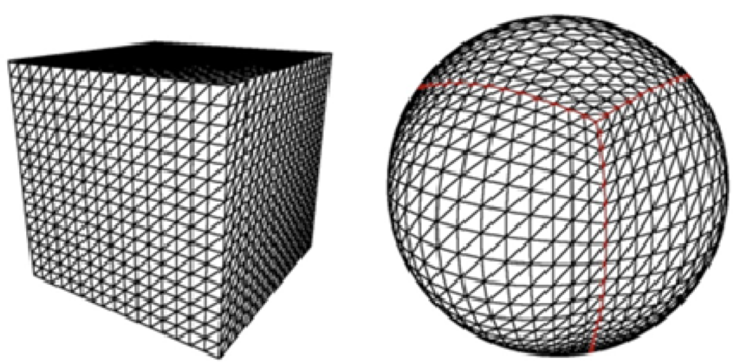
The `sphere_latlong` routine should form triangles in order to generate a sphere primitive. You need to specify discrete steps in both the elevation and azimuthal angles. Let the number of divisions for each angle be integers specified in the ImGui input controls for this primitive.





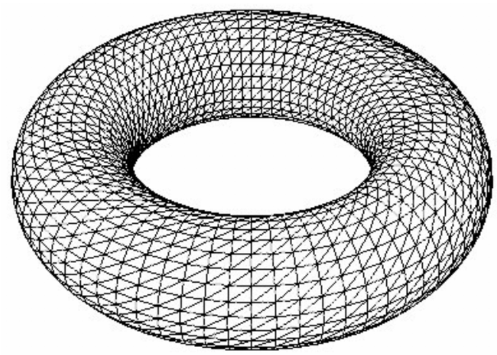
Completion (2 points)

The `sphere_from_cube` routine should start with a cube and divide each face into n by n squares and then moving the new vertices so that they lie on the sphere. Let n be an integer specified in the ImGui input control, i.e. if n is 1 your sphere is approximated as a cube.



Challenge (1 point)

The `torus_latlong` routine should form triangles in order to generate a torus primitive. You need to specify discrete steps in both the elevation and azimuthal angles. Let the number of divisions for each angle be integers specified in the ImGui input controls for this primitive.



Part 2: Shaders (8 points total)

Core (4 points)

Modify res/shaders/, add new vertex and fragment shaders that implement Cook-Torrance **(2pts)** and Oren-Nayar **(2pts)** light reflection models and apply them to the generated sphere (consider a point light source). Scale the result so that it does not look too bright.

Completion (2 points)

Write a routine that reads in an image (you are provided with Texture.png, NormalMap.png samples). Modify res/shaders/, add new vertex and fragment shaders that:

- map the image as a texture to the generated sphere **(1pts)**
- use the image RGB color values to perform normal mapping **(1pts)**

Challenge (2 points)

Implement the basic shadow mapping algorithm in two passes.

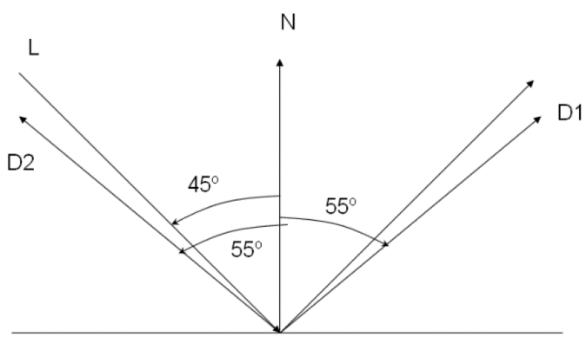
- Firstly, render the scene from the point of view of the light (consider directional lights only). Implement vertex and fragment shaders that compute and record the depth of each fragment **(1pts)**.
- Secondly, use the depth map to test whether the fragment that you computed is "behind" the shadow map and produce the final image with shadows **(1pts)**.

Part 3: Writing (5 points total)

Questions (3 points)

Type the answers or scan an image of your handwritten answers.

1. A glass sphere of radius 1 is sitting in clear air at the center of a coordinate system (i.e. the sphere center is at 0,0,0). A ray starting from point (5,5,4) in the direction of the point (0,0,.5) intersects the sphere. The index of refraction for the glass is 1.5. What is the direction of the ray reflected from the intersection point **(1pts)**?



2. Consider a surface with diffuse reflectance Kd = 0.1 and specular reflectance Ks= 0.5. Using the simple shading model described in class, find the exponent for the Phong specular reflectance such that for light incident from a direction L of 45 degrees to the surface normal the intensity of light reflected in the direction D1 that is 55 degrees from the surface normal is twice the intensity of the light that is reflected in direction D2, also 55 degrees from the surface normal, as shown. L, the surface normal, D1 and D2 all lie in the same plane. The surface does not emit light itself, and the rest of the environment is black. For the same case, find the exponent for the Blinn-Phong variant of Phong reflectance **(2pts)**.

Report (2 points)

Submit a 1-2-page PDF document outlining your experiences. In this report, you should include:

- Brief introduction of your functions in your programs, including how to run your program to get the desired results.
- The results of core, completion and challenge (depending on how much you have done) including images produced by your programs.