CSE 333/533 - Monsoon 2024
Assignment 4: Raytracing
Due date: 11:59:59, 30 Oct. 2024

In this assignment you will modify a raytracer to generate nice images. Lumina is a basic raytracer provided to you. Compile it and run to see the output. Currently it doesn't do much except rendering a sphere in flat color. Go through the code and understand how it works. Following questions ask you extend the raytracer to perform basic shading. For each of these, you are required to show images generated by the raytracer in your report.

1. Extend the Object class to implement a Triangle that can be rendered in the raytracer.

[Functionality: 9 marks, Code quality and doc: 1 mark, Total: **10 marks**]

2. Implement Blinn-Phong shading for the shapes.

[Functionality: 9 marks, Code quality and doc: 1 mark, Total: **10 marks**]

3. Implement shadows in the raytracer.

[Functionality: 9 marks, Code quality and doc: 1 mark, Total: **10 marks**]

4. Implement reflective and dielectric materials (i.e., make your ray-tracer recursive). This will include mirror reflection, simultaneous reflection and reflection at transparent surfaces, and total internal reflection.

[Functionality: 18 marks, Code quality and doc: 2 marks, Total: **20 marks**]

Deliverables (as a single zipped file **Assignment04_<studentID>.zip)** containing:

- C/C++ code (make sure to upload full code and do not include any intermediate object files, delete any other temporary files).
- A PDF Report written with with **Latex/MS Word**. Use the acmlarge option (single column) (see sample-acmlarge.tex if writing with Latex). Include screenshots within the report itself (and DO NOT attach separately).

Total marks for this assignment: 50 marks

Bonus *(bonus marks to a **maximum of 15** will be awarded for one or more of the following features. This part is completely optional)*

5. The generated image from this raytracer looks very jaggy. Implement jittered supersampling to improve the quality of your rendering. Perform the same only around edges to ensure performance. [5 marks]

[Functionality: (a) 4 marks, Code quality and doc: 1 mark, Total: **5 marks**]

6. Implement transformed primitives.

[Functionality: 4 marks, Code quality and doc: 1 mark, Total: **5 marks**]

7. Implement Beer's law to make the transparent objects look realistic.

[Functionality: (a) 4 marks, Code quality and doc: 1 mark, Total: **5 marks**]

Note: Your code should be written by you and be easy to read. You are NOT permitted to use any code that is not written by you. (Any code provided by the TA can be used with proper credits within your program and report)