

COSC 363: Computer Graphics

Assignment 2

Max. Marks: 20

Due: 11:55pm, 2 June 2023

Aim

In this assignment you will implement a ray tracer that can handle different types of geometric objects and global illumination features, and demonstrate its capability in enhancing the visual realism of a rendered scene.

1. The Basic Ray Tracer (Max. marks: 10)

In labs 6, 7, you will develop a simple ray tracer that can handle scenes containing planes and spheres. You will also implement methods for generating shadows and reflections. This assignment builds upon that ray tracer. As a minimum, your ray tracer should include the following features/objects.

- (a) A set of scene objects in a good spatial arrangement inside a box defined by five axis-aligned planes (“left”, “right”, “floor”, “ceiling”, “far”).
- (b) At least one transparent object such as a hollow sphere. (Note: A transparent object should not be modelled as a special case of a refractive surface with $\eta_1 = \eta_2 = 1$.)
- (c) Objects must cast shadows. Transparent and refractive objects must have lighter shadows.
- (d) The scene must include at least one planar object containing mirror-like reflections of surrounding objects.
- (e) At least one planar surface in the scene must have a chequered pattern. The method used for generating a stripe pattern in Lab-7 may be extended for this.

OpenGL functions should be used only for the final rendering of the ray traced image.

2. Extensions (Max. marks: 7)

This assignment will be marked out of 20, with 17 marks for the ray tracer and 3 marks for a report (see below). With just the basic features listed in Section 1, your ray tracer would earn at most 10 marks out of 17, and the report 2 marks out of 3. To get more than that, you need to implement a few additional features. Some possible features, and the approximate marks they would each gain if implemented correctly (up to a maximum of 7 marks for extra features) are given below.

- Objects other than a plane or a sphere: single/double cone (1.5 marks), cylinder (1 mark), torus (2-3 marks).
- A clearly visible “cap” for a cylinder or a double cone: 0.5 mark
- Refraction of light through an object (e.g. a refractive sphere) : 1 mark
- Multiple light sources including multiple shadows and multiple specular highlights generated by them: 1 mark
- Multiple reflections generated using parallel mirror-like surfaces: 1 mark

- Spotlight: 0.5 mark
- Basic anti-aliasing: 1 mark, Adaptive anti-aliasing: 1.5 marks. The report must contain outputs of the ray tracer with and without anti-aliasing.
- Textured sphere: 1 mark. Textured cylinder: 1 mark.
- A procedural pattern generated on any surface. The pattern must have a more complex structure than simple stripes or checks: 1 mark.
- Fog: 1 mark. The report must contain outputs of the ray tracer with and without the fog effect.
- Depth of field: 1-2 marks. The report must contain outputs of the ray tracer with and without the depth-of-field feature.
- Soft shadows: 1 mark

The marks associated with each feature should be taken to be indicative of the time and/or effort required to implement that a feature. The 10 marks for the minimum requirements are relatively easy to get when compared to the marks gained for some of the extension features. The above list should not be taken as a list of the only features that can be implemented. Marks will be given to only those features that are clearly visible in the ray traced output.

3. Report (Max. marks: 3, No. of pages: 2 – 4)

The report should describe your ray tracer, including both its successes and its failures. It should include at least one image showing the ray tracer's capabilities. You may include additional images demonstrating various features of the ray traced scene. The report should clearly outline both the mathematical and the implementation aspects of all extra features you have implemented in the ray tracer, other than those listed as minimum requirements. For example, if you have implemented anti-aliasing in your ray tracer, you should describe its purpose, the effect it produces on the output, and how it is implemented. Similarly, if you have used procedural textures, provide the mathematical equations used for mapping coordinates to colour values.

You must also give an estimate of the time taken by your program to generate the output on your personal computer or lab machine. The time taken to ray trace an image using a simple "brute force" algorithm is proportional to the number of pixels times the number of scene objects. To keep run times to a minimum, you should do most of your development with small images (approx. 600x600 pixels), and low levels of recursion.

Use of any code segments, data or images from external resources should be acknowledged in the report. Please include the details of the build process. You may also (optionally) include a list of references.

The report must contain the following declaration. **Assignment submissions without the declaration will not be accepted.**

<p style="text-align: center;">Declaration</p> <p>I declare that this assignment submission represents my own work (except for allowed material provided in the course), and that ideas or extracts from other sources are properly acknowledged in the report. I have not allowed anyone to copy my work with the intention of passing it off as their own work.</p> <p>Name</p> <p>Student ID</p> <p>Date</p>

4. Assignment Submission

Submit electronically (using *Learn*), the source code and any other supplementary files needed to run the program. Please also submit your report in PDF format. The files may all be packaged together and submitted as a single .zip file.

5. Miscellaneous

1. Check regularly on the *Learn* system forums for spec updates and clarifications.
2. This is not a group project. Each student must individually develop the program. In particular, students are not permitted to share program source code in any way. Standard departmental regulations regarding dishonest practices apply.