Computer Graphics Fall 2024

Farhad Kamangar kamangar@uta.edu

Assignment 06 Optional (Due date Nov. 24 2024)

Computer Graphics

Assignment 06 (Optional)

DUE DATE: Nov. 24, 2024

Purpose: Practice in Raytracing.

This is an optional assignment. If you choose to submit it, the grade for this assignment will replace the lowest assignment grade in the course.

Important Note:

This assignment will be graded on an all-or-nothing basis—partial credit is not available. All specified requirements must be met precisely, and the program must execute completely without any missing components for you to receive credit.

Assignment Details

Modify the program from assignment_05 to use the ray tracing algorithm to render an image.

On startup, the program should automatically read from the "cameras" file.

Commands to Implement:

Load (Open) Input File: Similar to assignment_05, but the output should now be displayed as ray-traced images.

Set the resolution: Allow the user to set the horizontal and vertical resolutions for the ray-traced images.

Set Light position: Allow the user to set the position of the light. default value should be (5.0,5.0,5.0).

Additional Notes:

- The formats of the cameras and data files are identical to those in assignment_05.
- The input file may contain data for both Bezier patches and polygon vertices.
- Ensure that cubic patches in the scene are rendered using ray tracing.
- Ensure that polygons in the scene are rendered using ray tracing.
- The program must use numpy for vectorized computations. You can use the ray tracing template provided as a starting point. Any matrix or vector operations, especially those involving pixel and ray computations MUST use numpy arrays.

Grading Policy:

- Since this assignment is optional, all features must work perfectly to receive credit.
- Partial credit will not be awarded. This means that both the procedure and final output must meet the requirements. For example, each step in the ray tracing algorithm—such as calculating intersections, handling lighting, and producing reflections—should function correctly.

⌃ Top

Computer Science and Engineering