CPSC 453 Fall 2019 Assignment 3

Introduction

This program implements a simple ray tracer that supports

- Spheres
- Triangles
- Triangle Meshes
- Shadows
- Phong lighting models
- Reflections (by recursively calculate the geometry's axis-aligned bounding-box)
- Loading simple .obj
- Refractions (by recursively calculate the geometry's axis-aligned bounding-box)
- Textured surfaces and the textures are read from .tga files
- Antialiasing through supersampling
- Rendering with the depth of field

Install

- Initiate in a Linux terminal and change the current working directory to the root of ***Assignment3***
- Execute the following instructions to compile

```
cmake .
make
```

Arguments

- The user can switch the rendering scene by providing arguments from standard input.
- To show the usage, please run by

```
./assignment3.out --help
```

- For rendering different scene, please run by

```
./assignment3.out sceneNumber
```

- A usage will be shown if any invalid arguments provided.

Uninstall

You may clean the temporary output files by running

```
rm -rf CMakeCache.txt
rm -rf ./CMakeFiles/
rm -rf Makefile
rm -rf assignment3.out
rm -rf cmake_install.cmake
rm -rf ./Output/*.ppm
```

Usage

- The output format is .ppm, you can open it directly by using GIMP in MS239

Highlight of each scene

- Since .ppm files cannot be shown in markdown, you can check ./Output where there are 4 scenes already

scene1.ppm

- A tetrahedron consists of triangles
- Two phong lighting spheres with different materials
- Two glass spheres with different materials
- Shadows

scene2.ppm

- Textures
- A big glass sphere in the middle
- Change some colors from the balls from previous scene

scene3.ppm

- Depth of field and it focus on the yellow ball
- The position of the yellow ball is changed, other factors are the same from the previous scene

scene4.ppm

- A teapot from the .obj file from last assignment is loaded into the middle of the scene
- No interpolated normal vectors for the teapot, I just use the normal of each triangle among the mesh

scene1_no_supersampling.ppm

- A no antialiasing through supersampling version of scene1.ppm, you can observe it from the edge of that tetrahedron.

About the code

- All data structures are written in C-style structs, all fields and member functions are exposed, only limited OOP stuff are used.
- It includes a simple .tga parser in the struct 'Texture' constructor and it uses bi-linear filtering as well.
- It includes a simple .obj loader in the struct 'TriangleMesh'.
- I referred some implementation of algorithms about 'Spatial Subdivisions', 'Hierarchical Bounding Volumes' and 'Beer–Lambert law' from https://www.flipcode.com/archives/ https://www.flipcode.com/archives/A%20faster%20voxel%20traversal%20algorithm%20for%20ray%20tracing.pdf, especially the grid traversal algorithm, honestly they are not easy to understand completely, so I follow some coding structure of them.
- Textures are downloaded from www.turbosquid.com and they can used for education purpose under its free license.
- I rewrote all functions and structs used as a 3D math library in Utility.hpp, as well as some useful functions.