Path tracer project plan - ELEC-A7151

Yuting Xie - 100878187

Aleksi Kääriäinen - 728971

Jukka Aho - 100496022

1. Project Scope

The goal of this project is to produce a software that renders a 3 dimensional image that simulates real-world illumination. The software will take a file describing a scene as input. After reading the input file, the program will render the scene and output a file on exit. The output file is a standard image format, such as ppm, png or jpeg.

This software will require one input file that describes the environment of the scene including location, angle of a camera and a list of different light sources and a list of geometries. Triangle, plane, and sphere are the three accessible geometries. Another input format .obj is also acceptable for a single geometry. Additionally, users can select the RGB color of the geometries as well as several materials, including diffuse, specular, and transparent materials.

The Monte Carlo method will be used to simulate three types of light sources: point light, area light, and global lighting. Furthermore, parallelized rendering on all CPU cores will be implemented for performance optimization.

In the end, a scene will be taken from the camera view and export an image. Following parameters for the camera could be customized: focus distance, field of view.

2. Communication

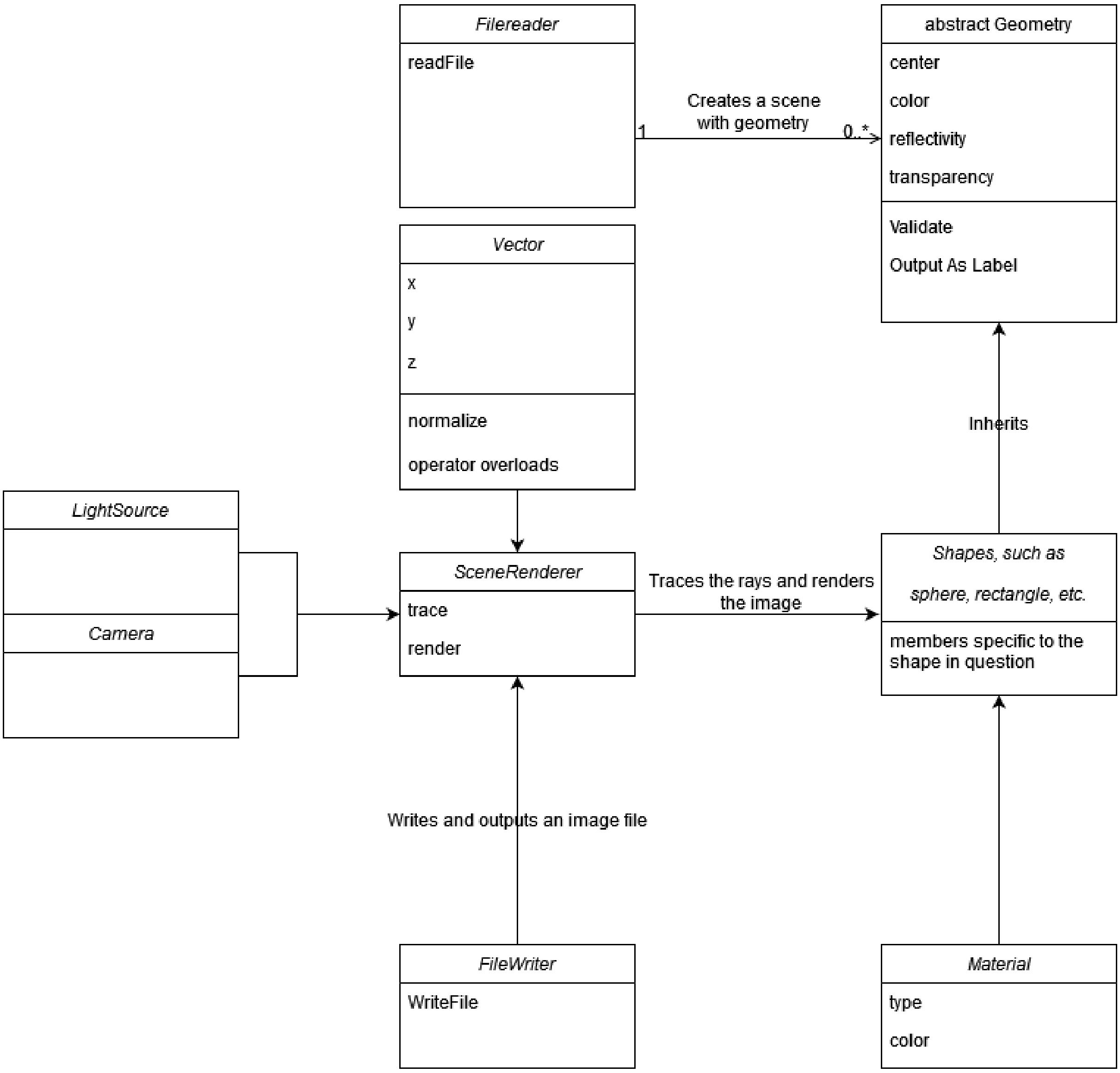
Telegram channel

3. Meeting times

Kick-off 8.11.2022 14:00

Weekly meetings: Tuesdays at 2 pm on Microsoft Teams

4. Class structure



The software consists of three main parts. The first part is the file reader, which will construct the scene from the input file. The input file is a text file that describes the scene. The scene itself consists of different geometries that can be made of various materials, which affect the light refraction off of the surface of the geometry. The most important part of the software is the scene renderer, which traces simulated light rays that are shot from the camera to the objects. After tracing the rays and calculating the color for every pixel in the scene, the scene will be passed to the file writer. The file writer writes and outputs the file in .ppm format.

5. External Libraries

- □ CMake
- □ Eigen
- □ libpng
- □ Google Test for unit testing
- □ Doxygen for documentation
- □ clang-format for code formatting
- □ SFML

C++ 17, GCC 9.4.0 x86

This project will utilize external libraries, even though the whole project could be implemented using only the standard libraries. Eigen provides a vast linear algebra library, which will be utilized in the tracing algorithm.

6. Schedule

Demo:

	Yuting Xie		**Aleksi Kääriäinen**		**Jukka Aho**	
08.11-11.11	Project plan	4h	Project plan	4h	Project plan	4h
14.11-18.11	Light source: point light, area light, global illumination	20h	Vector, Geometry, write file read file	20h	CMake Camera Simple path tracer	20h

21.11-25.11	Monte Carlo integration	20h	read file .obj	10h	Modifiable camera, Parallelized rendering on all CPU Cores	16h
28.11-02.12	diffuse, specular, and transparent materials	20h	Testing	20h	Parallelized rendering on all CPU Cores	10h

Final adjustment:

05.12-09.12	Debug	10h	Debug	10h	Debug	10h