Advanced Graphics 2016/2017 – Assignment 1

Introduction

This assignment is a preparation for assignment 2, which involves the construction of a real-time ray tracer. For assignment 1, you will be laying the ground work for this: a test bed for ray tracing. This test bed consists of the ingredients commonly required for ray tracing, plus a basic Whitted-style ray tracer.

Architecture

Ray tracing renders an image based on a scene description, given a camera and a screen plane, using rays originating from the camera and extending through pixels on the screen plane. The desired architecture implements ingredients for this: various classes (camera, screen, ray, light, primitive, …) and methods (creating a normalized ray, intersecting the ray with the scene, plotting pixels, … ). For subsequent assignments, we will extend and optimize this functionality. A solid test bed allows you to work on isolated parts of the framework without changing other parts.

Whitted-style Ray Tracing

To demonstrate the suitability of your framework, implement a Whitted-style ray tracer. This involves the construction of a ‘renderer’ class, which uses the other components to create an image. The Whitted-style ray tracer must at least support shadows and reflections.

Practical Details

The deadline for this assignment is December 6th, 23.59. You can hand in your assignment using the Submit system. The materials to submit are:

- your project, including sources and build instructions (if these are not obvious);
- a brief report, detailing implemented functionality, division of work, references and other information relevant to grading your submission.

You may work on this assignment alone, or with one other student.

You may implement the platform in C++ or C# or any other programming language that you may prefer. Note that support may be limited if you chose an alternative programming language. You may also target other operating systems than Windows, but again, support may be limited.

Feel free to discuss practical details on Slack. You are not supposed to share complete ray tracers there, but if everyone uses the same optimal ray/triangle test, that would be considered ‘research’.

Tasks & Grading

A passing grade (6) for this assignment requires:

- implementing a generic and extendible architecture for a ray tracer;
- a ‘free camera’ with configurable position, orientation, FOV and aspect ratio;
- a basic UI or controls to control the camera at run-time;
- support for at least planes and spheres;
- a basic material class;
- a basic scene consisting of a small set of these primitives;
- a Whitted-style ray tracing renderer to demonstrate and test the architecture.
To obtain additional points, you may work on the following:

1. Support for triangle meshes, e.g. using ‘obj’ files to import scenes.
2. Support for complex primitives.
3. Texturing.
4. Flexible lights: point lights, spot lights, IES profiles.
5. Refraction and absorption.
7. Performance tuning.

Please focus on features that are generic and applicable to any raytracing-based rendering algorithm.

**Purpose**

We will use the result of this assignment in the second assignment. The code you produce should therefore be reusable.

May the Light be with you,

- Jacco.