

**Proseminar Advanced Computer Graphics  
Summer Semester 2023**

***Programming Assignment Sheet 2***

**Hand-out: April 24**

**Hand-in: June 4**

**Path Tracing**

**Outline**

In this exercise, another implementation of an advanced computer graphics method will be extended. The provided program (*Example\_2.zip*) demonstrates global illumination rendering based on the path tracing method. The integral in the rendering equation is approximated via Monte-Carlo integration. Explicit direct lighting is included to improve image quality. The rendered view is saved in a file in PPM format.

The main target of the assignment is to get acquainted with advanced rendering computations, to implement theoretical concepts taught in the lecture, and to test the resulting extension.

**Tasks**

The exercise consists of extending the provided code by adding additional visual effects to the rendering process. The triangle intersection routines implemented in the previous assignment shall be integrated to be able to include more complex geometries. The changes should be made directly in the provided framework, without using any external libraries – with the exception of a possible loader for triangle meshes.

1. Extend the provided code to also handle objects represented as triangle meshes. For this, include the previously developed triangle intersection tests. For loading mesh geometries, feel free to add an external triangle mesh loader.
2. Extend the camera representation in the code to model a thin lens, thus including depth of field effects. A user should be able to provide values for aperture and focal length.
3. Extend the code to model glossy and translucent materials (i.e. going beyond perfectly reflective or transmissive materials only).

Submission of your solution is due on June 4, 2023 at 11:59 pm. Please submit the source code (i.e. no executables) via OLAT (submission access will be made available soon). You will then present your solution in front of the class during the proseminar on June 5.

The solution should be carried out in teams of 2 students. Please get in touch if you are unable to find a team. Make sure to acknowledge all team members in the code and submission (please name the ZIP file with all surnames). Only one submission per team is required.

Finally, please respect the academic honor code. Each team should work on its own on the solution of the assignment. Any attempts at plagiarism or collusion will lead to 0 marks and further scrutiny. Please get in touch in case of questions or problems with regard to the assignment.

In total there are 15 marks achievable in this assignment.

### **Implementation Remarks**

The solution should be developed using the provided framework. While you are free to develop in any environment, the final submission should compile and run in Linux or Windows. As reference, the machines in RR21 will be considered. The solution should compile using the provided make or CMake files. Follow a proper coding style and provide appropriate comments in the code, where extensions were made.