EXERCISES 2, 3 and ASSIGMENT 2

Exercise 2: March 26th

Exercise 3: April 2nd and 9th

Assignment 2: April 23rd

In assignment 2, you will implement the techniques of Distribution Ray Tracing covered in theoretical classes. Students will also extend their ray tracing application by using a grid-based acceleration structure.

Assignment 2 will build on top of exercises 2 and 3. Students' progress in those Exercises will be monitored in three laboratory classes (March 26th – April 9th). The goals for each of the exercises are described below.

ASSIGNMENT 2

Regarding stochastic sampling techniques, students need to implement:

- **Anti-aliasing** based on the Monte Carlo Sampling method (adaptive recursive supersampling) or the jittered method (2.5 pts)
- **Soft shadows** using an area of light with a set of N light source points and the random method (2.5 pts)
- **Depth of field** effect where the lens is simulated by a random distribution of N samples on unit squares or unit disks (3.5 pts)

Regarding the **acceleration structure** (8.5 pts), students should build a uniform grid. The grid traversal should use the Amanatides and Woo (1987) algorithm- http://www.cse.yorku.ca/~amana/research/grid.pdf. This paper is available in the Fénix Course page.

The application should allow the activation and deactivation of the grid in order to compare rendering performance.

Students should create **NFF scenes** (or other formats) (1 pts) that clearly demonstrate and leverage all previous techniques.

Students are encouraged to implement other techniques or functionalities (**extra**) that are not described in this document (2 pts).

EXERCISE 2

In Exercise 2, students will extend their ray tracing code by implementing the anti-aliasing technique based on the Monte Carlo Sampling method (adaptive recursive supersampling) or the jittered technique and Soft Shadows.

EXERCISE 3

In Exercise 3, students should end the previous tasks and implement the Depth of Field effect as well as to start the implementation of the uniform grid-based acceleration structure.

Lab Submission

Submit in the **Fenix** system your source code (.C and .h), and/or Makefile (if you have any) and a readme file specifying what are being submitted and how to compile and link your program.
All the files should be zipped in a file called Assignment 2.
Do not submit any executable files. We will use some sample NFF models to test your program.

ATTENTION: A **printed report with 6 pages** at maximum and a **short Making-Off Video** should be delivered by email until next Tuesday.

Late Penalty
You should submit your solution on time. Being late for one checkpoint could affect the time left for you to complete subsequent labs. All labs are due at the above specified due data, and there is a 20% penalty each day for up to 40%. After that, you get zero.

Grading Criteria
Grading of the labs will be based on the following:

- 90%: Correctness and adherence to assignment specification. Part of it will be checked on discussion and the demo provided by the Groups in the lab class regarding the checkpoint 2.
- 10%: Printed report, Video and readability, structure of code, use of comments, adherence to lab procedures (submitting, naming conventions, etc.)

Don't copy labs. Discussion of lab assignments is allowed and encouraged. However, you need to complete the lab all by yourself. Labs which are too similar will be properly handled by the teaching members of the discipline.