Dear student,

In this document I will describe your retake assignment for Advanced Graphics. **Please follow the instructions carefully.**

First of all: This is an **individual** assignment. That being said:

1. For this assignment, you start with a fully working path tracer. Please download it here.

The path tracer is written in C++ and should have a very familiar layout. Scene code is hidden in scene.cpp. Some helper functions are at the top of game.cpp. The scene uses Lambert diffuse materials, as well as pure speculars (mirror and dielectric). There is also a skydome, which is somewhat visible in the reflections. A single light source adds light to the scene.

The path tracer itself uses importance sampling (cosine weighted diffuse reflection) and next event estimation for the light source. **NOTE:** The implementation may or may not contain errors.

For this assignment, you will perform measurements, and you will make modifications to the path tracer. Your deliverables thus consist of a document and code, probably in several versions. Note that not every problem is entirely clear. **This is intentional.** Please use your knowledge of Advanced Graphics to carry out these tasks as well as possible.

2. To start, calculate a fully correct 'ground truth' image. Write down in as much detail as possible how you do this. Add the ground truth image itself.
3. Produce a graph of the RMSE ('root mean square error') for the original (unmodified) path tracer, for 1..1024 samples per pixel.

The path tracer currently has a recursion cap of 20 bounces.

4. Calculate the bias this cap introduces.

The path tracer uses NEE, which introduces a small error because of the estimation of the solid angle of the light source.

5. Calculate this error for a sequence of light sizes, starting at a light with a near-zero area, to a light that is four times larger than the one in the unmodified path tracer.

Some **ADVANCED** challenges:

The path tracer is currently not using multiple importance sampling.

6. Add multiple importance sampling and determine how this affects RMSE.

Replace the uniform random numbers that are currently used by blue noise, and carry out the following experiments:

7. Calculate the RMSE for the version with blue noise.
8. Calculate the RMSE for a 2x, 4x, 8x and 16x scaled down version of the original and the blue noise version and discuss your findings.

Please send your answers and code to bikker.j@gmail.com **on** (or before) **Friday April 23, 17:00**.

Have fun!