

# Report project 1, Advanced Graphics (2019)

Amber Elferink(5491525) & Ymke Wegereef (3971457)

## Implemented functionality

We have developed a render core using lighthouse 2 with the following functionality:

- correct handling of the 'ViewPyramid' handed to you by the RenderSystem
- support for triangles as handed to you via SetGeometry
- a basic but extendible material class
- a Whitted-style ray tracing renderer, running on the CPU, supporting shadows and reflections, to demonstrate and test the core

Furthermore, we have also implemented:

- A basic but extendible texture class
- read in .png pictures as Texture and display the according texture on triangles
- Loading the metallic setting from the CoreMaterial. Since we could not figure out a way to get a scene with correct translations with glass, we used another way to load the glass.
- Handling spot- and directional lights
- Area lights
- Dielectrics: Snell, Fresnel and Beer
- Efficient and generic multi-threading, yielding a 205% speedup on dual core with hyper threading.

## Division of the work

We have worked together on the first four (mandatory) points above. Once we had implemented a basic rendercore, Amber started working on the dielectrics and the multi-threading, while Ymke focussed on the different lights and the textures.

## References

- Our intersection method is adapted from the wikipedia page containing the Möller–Trumbore intersection algorithm:  
[https://en.wikipedia.org/wiki/M%C3%B6ller%E2%80%93Trumbore\\_intersection\\_algorithm](https://en.wikipedia.org/wiki/M%C3%B6ller%E2%80%93Trumbore_intersection_algorithm)
- For the dielectric and reflection algorithms, the slides were used as a guide.