1. **Personal details**

**Group number:** 106

**Members**

Rauno Arike -

Roland Bockholt -

Sem Hennekam - 4802578

1. **Work distribution**

The final work distribution is included in *final-project-workload-final.xlsx*. The final workload distribution hasn’t changed compared to the original planning.

1. **Standard features**

**3.1) Shading**

**3.2) Recursive ray-tracer**

**3.3) Hard shadows**

**General description**

Hard shadows are implemented in *testVisibilityLightSample(…)* in *light.cpp*. This function is called by *computeLightContribution(…)* in the same file, and should indicate if a light source should actually contribute to the light level of a place in the scene.

To achieve this, a ray is created from the light source (*samplePos*) to the point where the ray from the camera ends (*ray*), called *lightToIntersection*. The function then uses *bvh.intersect()* to find any intersections between *lightToIntersection* and any other triangle in the scene. If *bvh.intersect()* returns true, then *lightToIntersection* encountered another triangle, which means that the endpoint of the *ray* from the camera is in a shadow. Otherwise, *lightToIntersection* hasn’t.

**Rendered images**

To do: add rendered images with hard shadows, as soon as shading looks a bit more natural.

**Visual debugger**

Graphical user interface

Description automatically generated*To do: add another image from ray tracing*

**3.4) Area lights**

**3.5.1) Acceleration structure creation**

**3.5.2) Acceleration structure traversal**

**3.6) Barycentric coordinates**

**3.7) Textures**

1. **Extra features**

No additional features have been implemented

1. **Models**
2. **Performance test**