LEHRSTUHL GRAPHISCHE DATENVERARBEITUNG
Department Informatik

M. Stamminger, C. Weber, B.Keinert *Erlangen,* ***Monday, 12.01.2015***

Computer Graphics - Programming Exercises

Assignment 10 [6 Points] (Real-Time Raytracing)

In this assignment you will implement a real-time raytracer on the GPU. You only have to edit the fragment shader (rt_fs.glsl).

For this example we will not use any acceleration structures but perform brute-force intersection testing with all objects inside the scene. To get the final image a screen aligned quad is rendered. Rendering a screen aligned quad results in one fragment shader evaluation per visible pixel of the final image. Every fragment shader now has to send a ray into the scene and compute a color value that is presented to the viewer.

The scene is the same you already know from before. To visualize some nice effects that can be realized more easily with raytracing we added a "floor" to the scene. In general we will trade realism for cool effects! You can change back and forth between forward rendering and raytracing by pressing 'r'. The raytracing shader can be reloaded at runtime by pressing 'n'. There are additional instructions inside the source-code.

The following functions have to be implemented:

a) Ray-Sphere-Intersection
Inside the function interesectRaySphere you have to implement a ray sphere intersection. The function returns a struct called IntersectionTestResult. This struct contains a boolean if an intersection occured (isIntersection) and the distance of the intersection from the ray origin (tHit).

b) Ray-Scene-Intersection
Inside the function intersectRayScene you have to implement the intersection between a ray and the whole scene. This function returns a struct called SceneIntersectionTestResult, which contains a IntersectionTestResult of the closest intersection, the object ID of the intersected object (-1 if none), the normal at the intersection point and the intersection position in world space.

c) Shadow-Test
To get shadows you have to use a shadow test (already implemented in intersectRaySceneShadow) inside the getColor function. If the current hit point is inside a shadow multiply the color with 0.75.

d) Reflections
To realize mirror-like objects (sun and earth) we need additional secondary rays. These reflection rays have to be implemented at the end of the main-function.

Good Luck!

Your source code will be copied from your handin directory on:
Monday, 19.01.2015 14:00 pm
all subsequent changes cannot be taken into account!

Friedrich-Alexander-Universität
Erlangen-Nürnberg
TECHNISCHE FAKULTÄT