Praktikum 3: Hit Spheres

This third practical task is about equipping the raytracer with a ray-sphere intersection. Take a look at the new source code and understand how a ray is created pixel by pixel, from the origin of the viewer towards a virtual screen. In order to render photorealistic images, we first need to make sure that when a ray hits an object, we can determine the colour of the object that was hit. This time we will make it a little easier for ourselves. Good luck!

If you have any questions or problems, please write to us in Moodle!

Aufgabe 1 — Vectoren

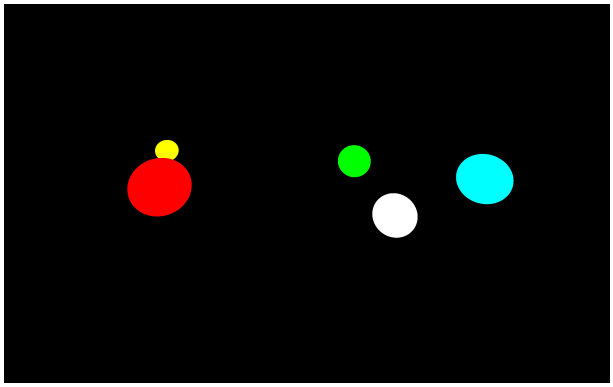

Figure 1: The result if the Ray-Sphere Intersection has been implemented correctly.

a) Open https://git.uni-due.de/vs.ude/objektorientierte-programmierung-cpp. Use git to *clone* the repository and import the project into 03_HitSphere/.

b) Adapt the source code so that the image from Figure 1 is achieved. In Ray.cpp you will find the `std::optional<Intersection> Ray::intersects(const Sphere& sphere) const {` method, which either returns an `Intersection` if a sphere was hit, otherwise only `{}`. Consider geometrically and mathematically how to find out if and where a line intersects a sphere. Hint: A straight line can intersect a sphere at 0 (no hit), 1 (tangent) or 2 points. Make sure that the front of 2 points is returned if the Ray shoots through the sphere.

c) Note on submission: The submission should be a .zip archive containing only C++ source code and a file showing who the group members are with first name, surname and matriculation number.