# Praktikum 3: Hit Spheres

In dieser dritten Praktikumsaufgabe geht es darum den Raytracer mit einer Ray-Sphere Intersection auszustatten. Schauen Sie sich den neuen Quelltext an und verstehen Sie wie grundsätzlich Pixel für Pixel ein Ray erstellt wird, von dem Ursprung des Betrachters in Richtung eines virtuellen Bildschirms. Damit photorealistische Bilder gerendert werden können, müssen wir zunächst dafür sorgen dass wenn ein Ray ein Objekt trifft, wir die Farbe des Objektes ermitteln können welches getroffen wurde. Das machen wir uns dieses mal etwas einfach. Viel Erfolg!

Schauen Sie sich dazu an wie operator overloading in C++ funktioniert.

Bei Fragen oder Problemen schreiben Sie gerne im Moodle!

## **Aufgabe 1** — Vectoren

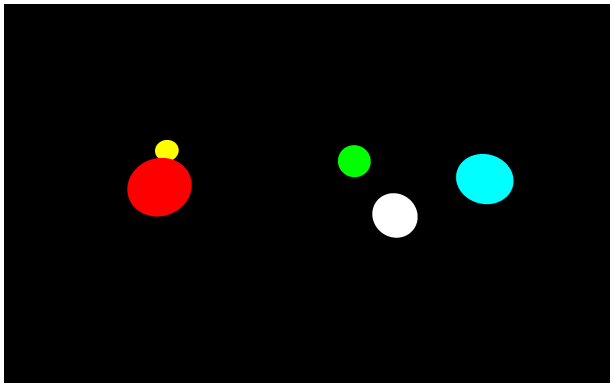

Figure 1: Das Ergebnis wenn die Ray-Sphere Intersection richtig implementiert wurde.

a) Öffnen Sie https://git.uni-due.de/vs.ude/objektorientierte-programmierung-cpp. Nutzen Sie git um das Repository zu *clonen* und importieren Sie das Projekt in 03_HitSphere/.

b) Passen Sie den Quelltext so an, damit das Bild aus Figure 1 erreicht ist. In Ray.cpp finden Sie die `std::optional<Intersection> Ray::intersects(const Sphere& sphere) const {` Methode, welche entweder eine `Intersection` zurückgibt wenn eine Sphäre getroffen wurde, ansonsten nur `{}`. Überlegen Sie geometrisch und mathematisch, wie sie herausfinden ob und wo eine Gerade eine Kugel schneidet. Hinweis: Eine Gerade kann eine Kugel in 0 (kein Treffer), 1 (Tangente) oder 2 Punkten schneiden. Achten Sie darauf dass der vordere von 2 Punkten zurückgegeben wird, falls der Ray durch die Sphäre schießt.

c) Hinweis zur Abgabe: Die Abgabe soll ein .zip Archiv sein, welches nur C++ Quelltext enthält und eine Datei aus der hervorgeht wer die Gruppenmitglieder sind mit Vorname, Nachname und Matrikelnummer.