Praktikum 2: Raycast

This second practical task is about implementing a vector class so that direction vectors are calculated and displayed correctly. Good luck!

Take a look at how operator overloading works in C++.

If you have any questions or problems, please post them in Moodle!

Aufgabe 1 — Vectors

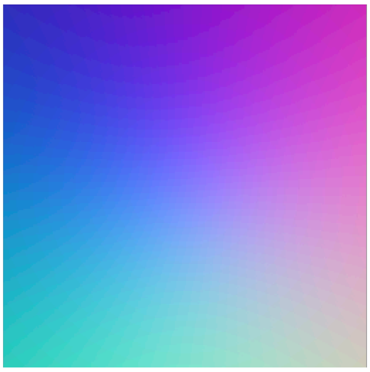

Figure 1: The result if the vector class has been implemented correctly.

a) Open https://git.uni-due.de/vs.ude/objektorientierte-programmierung-cpp. Use git to *clone* the repository and import the project into 02_Raycast/.

b) Adapt the source code so that the image from Figure 1 is achieved. To do this, you must fill the methods and functions in Vector3.cpp. Note: Even if not all methods are required to generate the image, you must implement all methods/functions in Vector3.cpp in order to pass the practical course.

c) Note on submission: The submission should be a .zip archive containing exactly 4 files. The main.cpp, Vector3.cpp, Vector3.hpp and a file showing who the group members are with first name, surname and matriculation number.