

Übung Nr. 4:

Das Programm `ObjElabGLaaTx.c` soll ein Lichtmodell erhalten, das auf die dargestellten Modelle angewandt wird (Abb. 1). Der Programm-Code (s. entsprechendes MS-VC-Projekt unter <http://homepages.thm.de/christ/>) ist dazu vorbereitet, es fehlen etwas über 10 Zeilen Code an Stellen, die mit `MORE_LITE` gekennzeichnet sind; diese sind anhand der u.a. Beschreibung zu ergänzen.

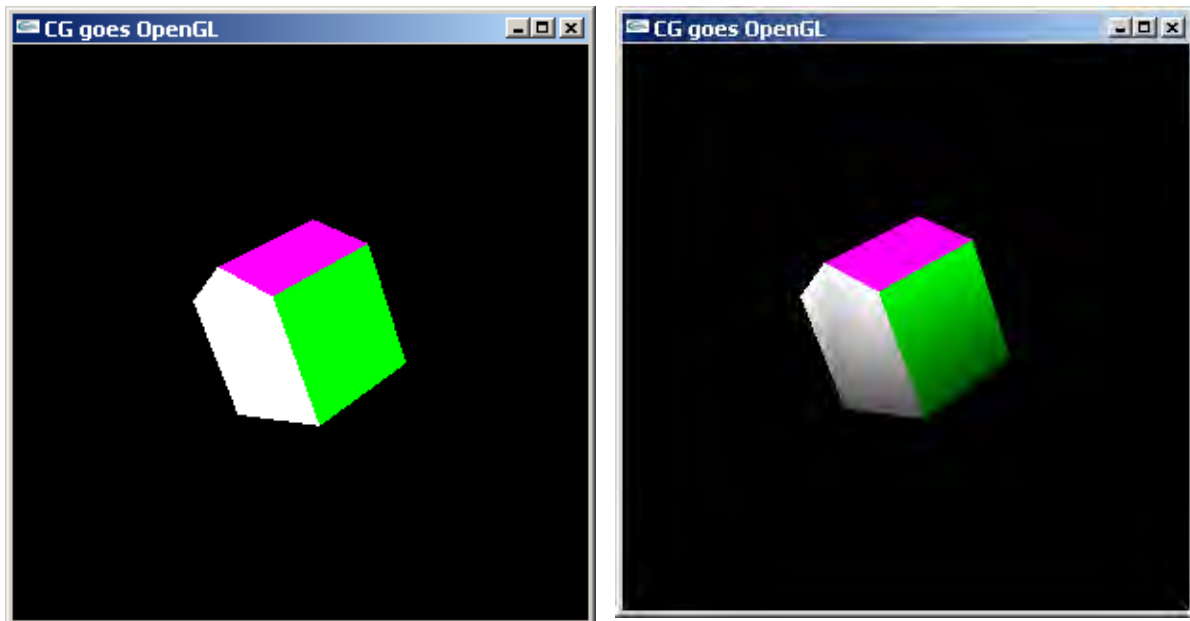


Abb. 1 Objekt ohne (li.) und mit (re.) simulierter Beleuchtung

Als Einstieg in die Problematik sollte der Code von `LitInit()` näher betrachtet werden; diese Funktion ist vollständig und bietet dadurch einen praxisbezogenen Überblick über die in der Vorlesung vorgestellten Inhalte. Interessant ist z.B., sich mit der Frage auseinanderzusetzen, woran die Art, die Lage und die Strahlungsrichtung der Lichtquelle zu erkennen ist.

Das heruntergeladene Programm läßt sich compilieren und starten. Bei Zuschaltung der Lichtquelle (Taste 'l') ist aber das dargestellte Modell kaum noch zu erkennen (Abb. 2 li.). Grund hierfür ist zunächst die noch nicht korrekte Ausrichtung der Normalen. Da Normalen in OpenGL ausschließlich Eckpunkten zugeordnet werden und „radial“ ausgerichtete (d.h. vom angenommenen Objekt-Mittelpunkt wegweisende) Normalen oft eingesetzt werden, um „weiche“ (kontrastarme, gerundete) Ecken und Kanten darzustellen, und da die hier verwendeten CGF-Objekte um ihren Koordinatenursprung modelliert sind, ist es hier ratsam, an der entsprechenden Stelle im Programm die Punkt-Koordinaten zugleich als Normalen-Koordinaten einzusetzen.

Nach Ausrichtung der punktbezogenen Normalen ist der Beleuchtungseffekt zu sehen, jedoch ohne die Farben, die zuvor den Objektflächen zugewiesen wurden (Abb. 2 re.). Ursache dafür ist wiederum, daß den Objekt-Flächen noch keine Material-Eigenschaften zugewiesen wurden: Sie reflektieren alle Komponenten des darauf fallenden Lichtes. Hier können die Farb-Komponenten der jeweiligen Fläche auch als ambiente, diffuse und spekulare Farbeigenschaft der jeweiligen Objektfläche eingesetzt werden. Das Ergebnis sollte das erwartete Bild des Objektes in der simulierten Beleuchtung sein (Abb. 1 re.).

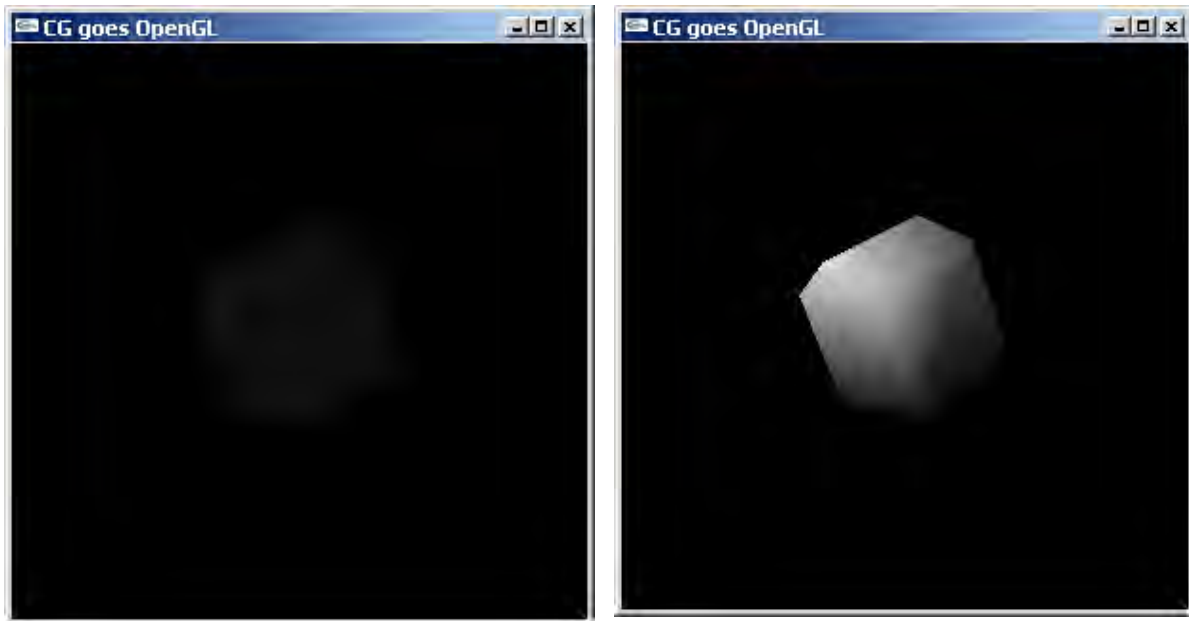


Abb. 2 Objekt vor (li.) und nach (re.) der Normalen-Ausrichtung

Als letztes sollte die Lichtquelle animiert werden; sie soll durch drücken der Taste 'L' um jeweils 30 Grad im positiven Drehsinn (was bedeutet das?) um die y-Achse gedreht werden. Dazu ist die Funktion `rotLitY()` eingerichtet worden, die auch entsprechende Berücksichtigung im Code findet (s. Funktionen `key()`, `LitInit()`). Die Funktion `rotLitY()` soll vervollständigt werden.

Da diese Teilaufgabe ein gutes Repetitorium für die eigenständige (unabhängige) Animation von Objekten darstellt, sollte hier das Gespräch zum Lehrenden gesucht werden und die wenigen (ca. 5) fehlenden Code-Zeilen nicht durch Ausprobieren und Raten gefunden werden.

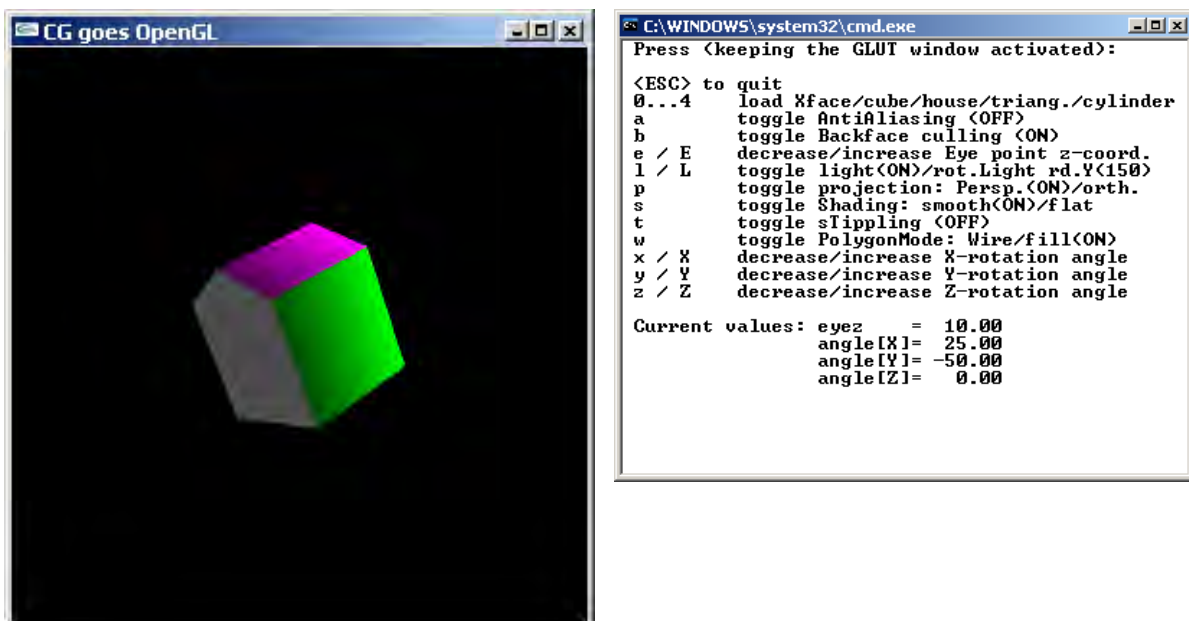


Abb. 3 Objekt nach Rotation der Lichtquelle um 150° (li.) und dazugehöriges Menü (re.)