

**Übung Nr. 3:**

Geladene dreidimensionale grafische Flächenmodelle sollen im Konsole-Fenster mit ASCII-Zeichen die Schattierungsart Flat Shading erhalten (Abb.1). Hierzu wird unter <http://homepages.thm.de/christ/> das MS-VC-Projekt WireCullFillShad bereitgestellt. Die gleichnamige C-Datei (WireCullFillShad.c im Projekt-Verzeichnis Src) ist nicht komplett, sie braucht aber auch nicht vervollständigt zu werden; denn die fehlenden Anweisungen ihrer Funktionen sind vorcompiliert in der Datei WireCullFillShad.obj enthalten, die anstelle der Quelle im Projekt eingebunden ist.

In ähnlicher Weise sind größere Code-Teile als \*.obj-Dateien dem Projekt hinzugefügt worden. Es handelt sich zumeist um mathematische und grafische Funktionen, die in den Übungen zum Bachelor-Teil dieses Faches behandelt und codiert wurden. Den im Projekt einsehbaren, kommentierten Code-Fragmenten (vor allem in den Dateien MatrOps.c und GrafOps.c im Verzeichnis \_Libs) kann sowohl die Anwendung als auch die Funktionsweise der entsprechenden C-Funktionen entnommen werden. Diese sind bestens dazu geeignet, die hier benötigten Funktionalitäten mit wenig Aufwand zusammenzustellen.

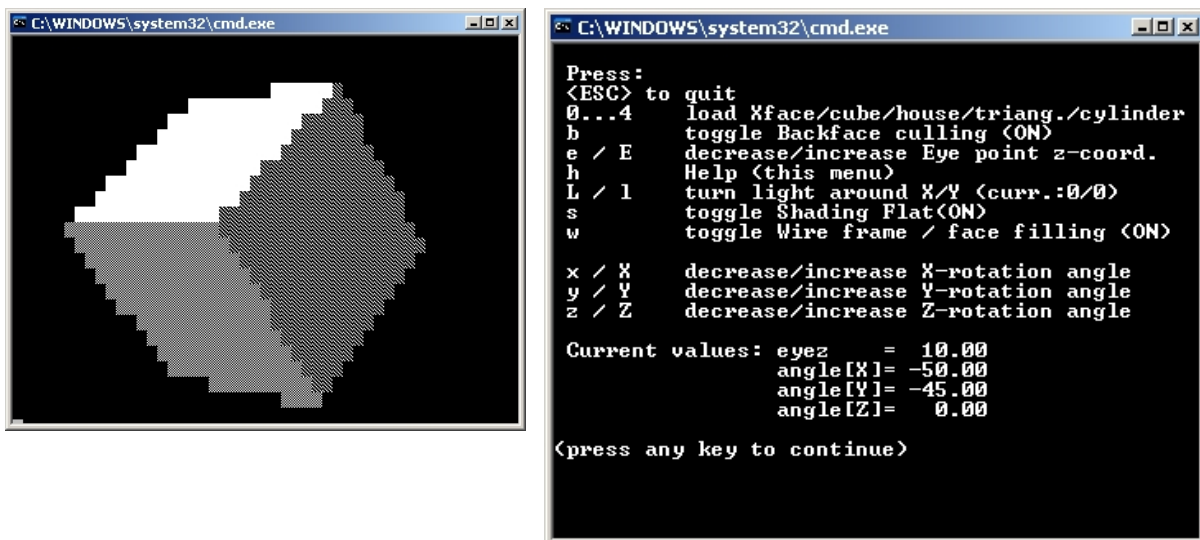


Abb. 1 WireCullFillShad.exe: Darstellung eines Würfel-Modells (li.) und dazugehörige Menü-Einstellungen (re.)

Die vorliegende Übung besteht lediglich in der Fertigstellung der zwei Funktionen in der Datei shadOps.c:

```
float reflex (float *light, float *normal, float ambient)
```

soll die Lichtstärke des diffus reflektierten Lichtes berechnen und zurückgeben.

In der hiesigen Aufgabe wird angenommen, die Beleuchtung habe die Stärke eins, und das ambiente Licht werde mit einem Betrag von 0.1 der diffus reflektierten Lichtstärke hinzuaddiert.

*Es ist außerordentlich hilfreich für die erfolgreiche Behandlung dieser Übung, festzustellen, an welchen Stellen im Code die obigen Angaben zu erkennen sind und in welcher Richtung die Lichtquelle strahlt.*

## Die Funktion

```
int turnRndXY (float *point, float *xyAngle)
```

dreht einen Punkt (hier: die Lichtquelle) um den Koordinaten-Ursprung nacheinander um den Winkel `xyAngle[X]` um die x-Achse und dann um den Winkel `xyAngle[Y]` entsprechend um die y-Achse (Abb. 2).

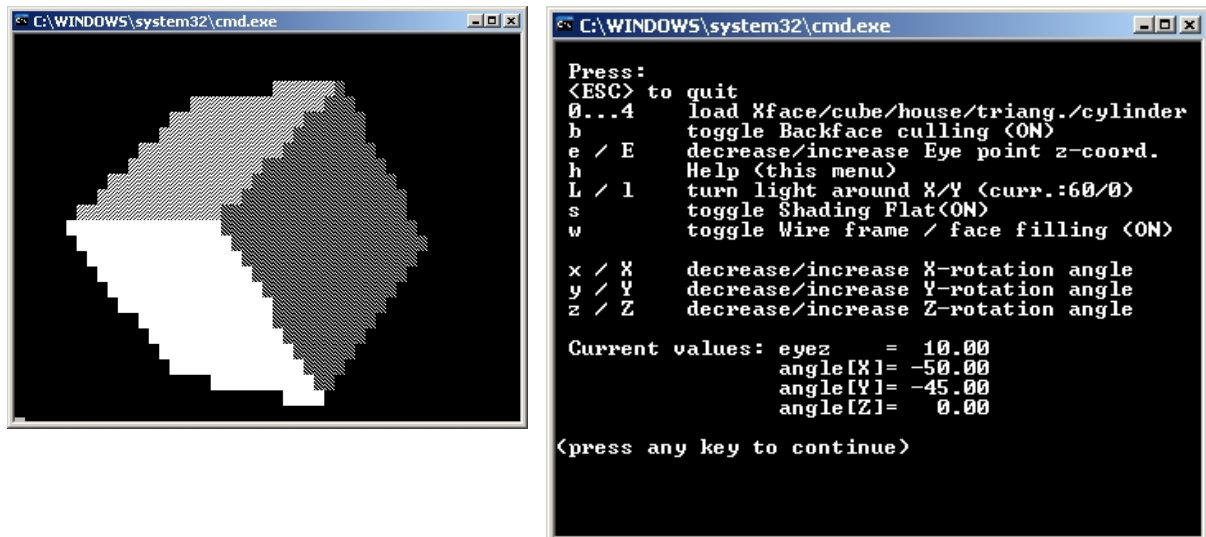


Abb. 2 `WireCullFillShad.exe`: Darstellung des Würfels nach Abb.1 nach Drehung der Lichtquelle um  $60^\circ$  um die x-Achse (li.) und dazugehörige Menü-Einstellungen (re.)

Auch hier ist, wie im o.a. ersten Teil dieser Übung, wichtig, sich zunächst bewußt zu machen, wie Position und Strahlungsrichtung der Lichtquelle dargestellt und behandelt werden, bevor mit der Codierung der wenigen fehlenden Anweisungen begonnen wird.

Eine weitere (nicht zur Bearbeitung gehörige) Einzelheit sollte noch beachtet werden:

Die Entscheidung über die Zuweisung eines Symbols (allg.: eines Farbwertes) jeder Fläche erfolgt durch die Routine `gLevel()` (Datei `interOps.c`, im o.a. MS-VC-Projekt Bestandteil von `lineNfillOps4shad.c`) und stellt eine lineare Abbildung der Form „Symbol = f (Reflexion)“ dar. In `gLevel()` kann allerdings die Entwicklungsarbeit zur Bresenham-gestützten Interpolation nicht unmittelbar genutzt werden; denn während dort eine lineare Beziehung zwischen zwei ganzzahligen Variablen umgesetzt wurde, handelt es sich hier zunächst um ganzzahlige Farbwerte, die anhand eines reell-wertigen Arguments (der Reflexion) linear interpoliert werden. Da dies nicht Gegenstand dieser Übung ist, wird die codierte Routine in `lineNfillOps4shad.obj` bereitgestellt.