BERGISCHE UNIVERSITÄT WUPPERTAL
FAKULTÄT FÜR MATHEMATIK
UND NATURWISSENSCHAFTEN

BERGISCHE
UNIVERSITÄT
WUPPERTAL

Angewandte Informatik
Dr. Martin Galgon
M.Sc. Jose Jimenez

Bildgenerierung

Wintersemester 2023 / 2024

Übungsblatt 5

Aufgabe 13 *(3D-Clipping für perspektivische Projektion)*

Die Projektion aus Aufgabe 10 enthielt noch einige Vereinfachungen. So wurde die z-Koordinate vernachlässigt und kein Clipping in normalisierten Sichtkoordinaten durchgeführt.

Das Rahmenprogramm `proj2.cc` im Verzeichnis `/home/bildgen/Aufgaben/projektion-2` enthält alle Schritte für den allgemeinen Fall. Ihre Aufgabe ist, das fehlende Clipping zu ergänzen. Sie sollen hierbei keine optimierten Verfahren wie Cohen-Sutherland oder Cyrus-Beck-Liang-Barsky auf den 3D-Fall übertragen. Gehen Sie einfach von den Parameterdarstellungen der Gerade aus, d. h.

$$x(t) = x_0 + t(x_1 - x_0), \quad y(t) = y_0 + t(y_1 - y_0), \quad z(t) = z_0 + t(z_1 - z_0), \quad 0 \le t \le 1\,.$$

Alle notwendigen Parameter und Daten sind im Code vorgegeben. Für jede Kante wird die Funktion `clip3D` aufgerufen. Die Endpunkte der unbearbeiteten Kante werden als Referenz via `anf` und `end` übergeben. Führen Sie nun für die jeweilige Kante *nacheinander* Clipping an den Clippingebenen in normalisierten Sichtkoordinaten aus. Aktualisieren Sie dann `anf` und `end` mit den neuen Endpunkten der geclippten Kante.

1. $-1 \le z \le z_{\min}$

   Bestimmen Sie die Schnittpositionen t_1, t_2 mit der Front- und Backplane aus $z(t_1) = -1$ und $z(t_2) = z_{\min}$. Beachten Sie auch den Sonderfall für Geraden, die in der x-y-Ebene liegen.

2. $z \le x \le -z$

   Bestimmen Sie t_1, t_2 aus $x(t_1) = z(t_1)$ und $x(t_2) = -z(t_2)$.

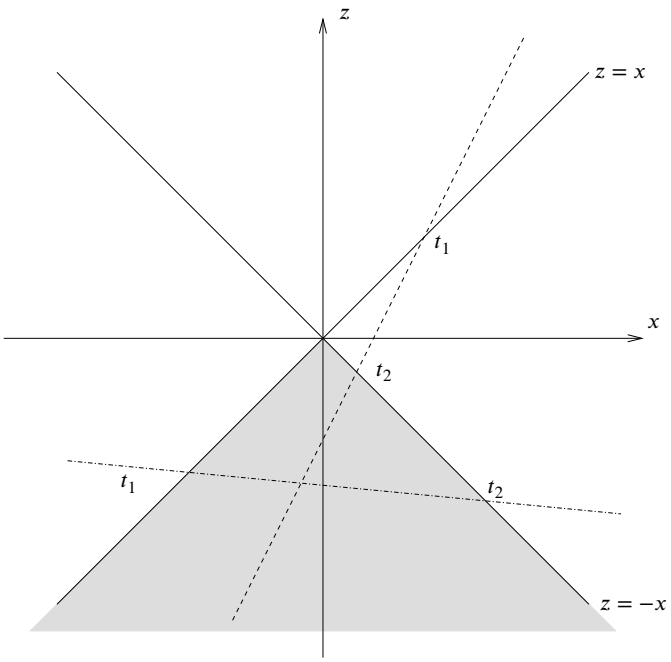
   Überlegen Sie sich, welche verschiedenen Lagen eine Kante und ihre Endpunkte haben kann, so dass sie die zulässige (graue) Fläche schneidet (siehe Illustration). Dokumentieren Sie die verschiedenen Fälle entweder in Ihrem Quelltext oder mit Hilfe separater Zeichnungen (oder beides). Beachten Sie hier den Sonderfall, dass die Gerade parallel zur y-Achse liegt.

3. $z \le y \le -z$

   Analog zu Fall 2.

Testen Sie das Programm mit den Dateien `Wuerfel?.info` (es ist jeweils kommentiert, welcher Teil ausgeblendet werden sollte), `Colosseum?.info` und `sts1.info` sowie den zugehörigen Modellen `Wuerfel.in`, `Colosseum.in` und `sts.in`. Hierzu müssen Sie Ihr Programm z.B. wie folgt aufrufen:

```
cat Wuerfel.in Wuerfel1.info | proj2
```



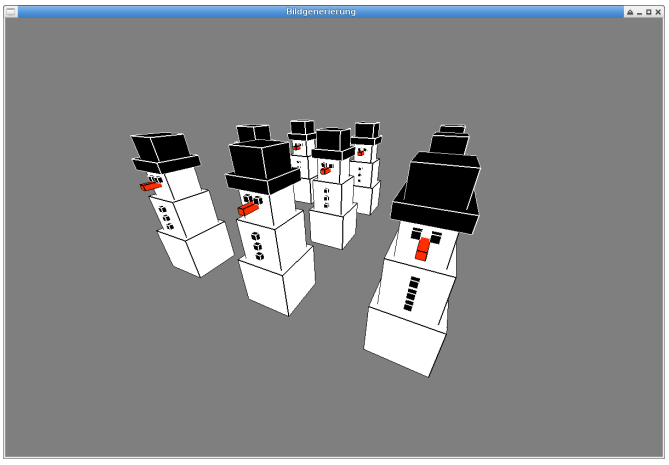
Aufgabe 14 *(Modellierung mit OpenGL)*

Im Verzeichnis `/home/bildgen/Aufgaben/opengl-1` finden Sie ein Rahmenprogramm zu dieser Aufgabe, das die Initialisierung von `OpenGL` für Sie übernimmt. Mittels eines Aufrufs von `make` können Sie es kompilieren.

Ergänzen Sie das Zeichnen einer Gruppe von Schneemännern. Vervollständigen Sie dafür die Funktionen `drawSnowman` und `drawSnowmen`.

Die einzelnen Teile der Schneemänner sollen aus Quadern bestehen. Dafür existiert bereits eine Funktion namens `drawCube(const DrawColour& col)`, die einen Würfel mit den Eckpunkten $(-1,-1,-1)$ und $(1,1,1)$ sowie der übergebenen Farbe zeichnet. Mit `glScalef` und `glTranslatef` können Sie die Würfel skalieren und verschieben. Bei der Skalierung kann man für jede Koordinatenrichtung einen eigenen Faktor angeben, so dass man den Würfel in einen Quader transformieren kann. Weiterhin werden Sie die Befehle `glPushMatrix` und `glPopMatrix` an geeigneten Stellen benötigen.

Unten sehen Sie eine Beispielausgabe einer Lösung dieser Aufgabe.



Abgabe: Mi., 29.11.2023, 13:15 Uhr

Senden Sie Ihre Lösungen der Theorie-Aufgaben und Ihre Programme per E-Mail an `bildgen@studs.math.uni-wuppertal.de` .