HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG
Fakultät für Informatik
Labor für Computergrafik
Prof. Dr. G. Umlauf

Konstanz, 26.09.2016

Übungsblatt 3

„Computergrafik“

Besprechung und Abgabe spätestens am 20.12.2016, F033.

Vorbemerkung: Benutzen Sie für diese Übung keine OpenGL, GLUT oder GLAUX Funktionen, die Projektionen und Rotationen berechnen! Benutzen Sie die zur Verfügung gestellten Vektor und Matrix Klassen.

Aufgabe 5 (Zentralprojektion) — 2+1+1+2 Punkte

Implementieren Sie eine Applikation, mit deren Hilfe die Zentralprojektion entlang der z-Achse einer Szene dargestellt werden kann. Gehen Sie dazu wie folgt vor:

a. Implementieren Sie eine Funktion

```
CVec4f projectZ(float fFocus, CVec4f pSicht)
```

zur Zentralprojektion eines beliebigen Punktes `pSicht` in homogenen Koordinaten auf die Projektionsebene. Gehen Sie davon aus, dass wie in Abbildung 1

- der Augpunkt auf der positiven z-Achse liegt, d.h. (`0,0,fFocus`),
- die Blickrichtung anti-parallel zur z-Achse verläuft,
- der View-Up-Vektor (y-Achse der Bildebene) parallel zur y-Achse ist, und
- die Bildebene die xy-Ebene ist

Die Fokallänge `fFocus` soll der Funktion als Parameter übergeben werden können.

b. Implementieren Sie eine Funktion

```
void drawProjektedZ (CVec3f Points[8]),
```

die bei Übergabe von acht 2D-Punkten in geeigneter Reihenfolge einen Quader als projiziertes Drahtgittermodell zeichnet. Dazu müssen Sie lediglich die entsprechenden projizierten Punkte mit Kanten verbinden. Benutzen Sie dazu den Bresenham-Algorithmus.

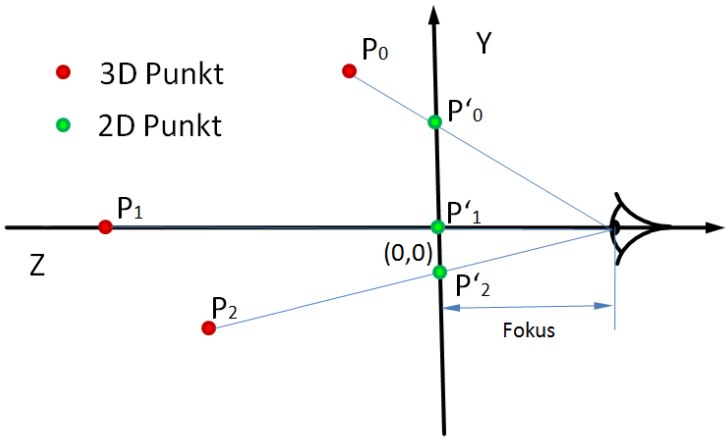

Abbildung 1 Zentralprojektion entlang der z-Achse.

c. Legen Sie 3D-Daten für mindestens drei sinnvoll verteilte Quader an. Da Sie pro Quader lediglich acht 3D Punkte speichern müssen, können Sie dies entweder mit Hilfe eines Arrays, einer Struktur oder einer einfachen Klasse realisieren.

d. Implementieren Sie eine Funktion

```
void drawQuader(CVec3f Quader[8],float fFocus, Color c),
```

- die als Argument einen Quader akzeptiert,
- die 3D-Punktdaten mit `projectZ(...)` auf die Bildebene projiziert und
- die die entsprechenden Verbindungslinien zeichnet.

Bauen Sie diese Funktion in Ihre `display`-Funktion ein, um Ihre Szene anzuzeigen.

Aufgabe 6 (Allgemeine Sichteinstellung) 7+1 Punkte

Bisher können Sie die Szene nur aus einer einzigen Ansicht betrachten. In dieser Aufgabe sollen eine allgemeine Sichttransformation implementieren werden. Gegeben seien also zusätzlich zu den Parametern aus Aufgabe 5:

- Eine allgemeine Position des Augpunktes (in homogenen Koordinaten) `EyePoint`.
- Eine allgemeine Sichtrichtung `ViewDir`.
- Ein allgemeiner View-Up-Vektor `ViewUp`.

Der Augpunkt und die beiden Vektoren implizieren dabei ein vollständiges dreidimensionales Koordinatensystem. Der fehlende Vektor für die x-Achse kann durch $x = y \times z$ berechnet werden. Gehen Sie dabei wie folgt vor:

a. Implementieren Sie eine Funktion

```
CMat4f getTransform(CVec4f viewOrigin, CVec4f viewDir,
                    CVec4f viewUp),
```

die die Gesamttransformation als 4×4-Matrix zur Transformation des Sicht-Koordinatensystems in das Welt-Koordinatensystem berechnet (siehe Abbildung 2). Die Inverse dieser Matrix überführt Punkte aus dem Welt-Koordinatensystem in das Sicht-Koordinatensystem.

b. Implementieren Sie eine Funktion

```
CVec4f projectZallg(CMat4f matTransf, float fFocus,
                    CVec4f pWelt),
```

die erst den übergebenen Punkt `pWelt` mit `matTransf` in das Sicht-Koordinatensystem überführt und dann mit der Funktion `projectZ` auf die Bildebene projiziert.

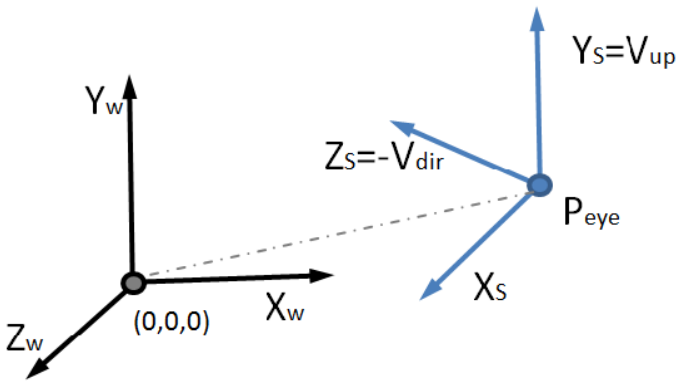

Abbildung 2 Allgemeine Sichttransformation.

HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG
Fakultät für Informatik
Labor für Computergrafik
Prof. Dr. G. Umlauf

Aufgabe 7 (Zusammenführung) 1+3+3+2+1 Punkte

Schreiben Sie nun eine Applikation, mit der die Szene aus Aufgabe 5 wahlfrei betrachtet werden kann. Mit Aufgabe 6 haben bereits Sie die Möglichkeit bei gegebenem Sicht-Koordinatensystem Ihre Szene allgemein zu betrachten. Für eine vollständige Applikation fehlt nun lediglich noch eine "komfortable" Möglichkeit das Sicht-Koordinatensystem einzustellen. Erstellen Sie dazu eine Keyboard-Interaktion mit den folgenden Tastenbelegungen:

a. F vergrößert die Fokallänge und f verkleinert die Fokallänge.

b. X, Y und Z rotieren das Sicht-Koordinatensystem in positiver Richtung um die x-, y- und z-Achse des Welt-Koordinatensystems und x, y und z drehen in negativer Richtung um die jeweilige Achse.

c. A, B und C bzw. a, b und c rotieren das Sicht-Koordinatensystem in entsprechender Richtung um die jeweilige Achse des Sicht-Koordinatensystems. (A, a: Sichtvektor, B, b: View-Up-Vektor, . . .).

d. U, V, W, u, v und w verschieben das Sicht-Koordinatensystem entlang der Achsen des Welt-Koordinatensystems in die entsprechenden Richtungen. (U, u: x-Achse, V, v: y-Achse, W, w: z-Achse).

e. R setzt das Sicht-Koordinatensystem in den Anfangszustand (deckungsgleich mit dem Welt-Koordinatensystem) zurück.

Zum Zeitpunkt der Initialisierung der Applikation sollen Welt-Koordinatensystem und Sicht-Koordinatensystem deckungsgleich sein.

Besprechung und Abgabe spätestens am 20.12.2016, F033.