

**Klausur
Computergrafik
für Bachelor-Studierende
SS 2018**

– Lösungshilfe –

Personalien:

Name, Vorname:

Matrikelnummer:

Ich möchte mein Ergebnis anhand meiner Matrikelnr. auf der Homepage zu diesem Fach für begrenzte Zeit abrufen können. (Bitte ggf. ankreuzen und unterschreiben)

(Unterschrift)

Hinweise:

- Die Bearbeitungszeit beträgt 90 Minuten.
- Alle schriftlichen Hilfsmittel sind zugelassen; andere Hilfsmittel, insb. elektr. Rechen- und Kommunikationsapparate, dürfen nicht verwendet werden.
- Ausgesprochene Folgefehler (durch Übertragung falscher Zwischenergebnisse) werden in Folgerechnungen als richtig gewertet.
- Die Aufgaben sollen nur auf diesen Blättern (inkl. Rückseite) bearbeitet werden. Bei Bedarf wird zusätzliches Papier zur Verfügung gestellt.
- Zur sicheren Zuordnung aller Lösungen wird um eine persönliche Kennung (Name u./o. Matrikelnr.) auf allen Blättern gebeten.
- Auf Wunsch darf auch Bleistift verwendet werden.
Zur leichteren Lesbarkeit werden Substantive nur in einem Geschlecht („Nutzerin“) verwendet.

1. **Aufgabe** (15 Punkte)

- a) Sie lernen jemanden kennen, der als Informatiker mit Erfahrung auf dem Gebiet der Datenbanken jahrzehntelang als Entwickler im Dienst der internationalen Flugsicherheit gearbeitet hat. Sie erfahren, daß er berühmt geworden ist, weil er ein Verfahren entwickelt hat, mit dem nach Einscannen des Paßbildes und Eintippen der Paßnummer sofort eine Liste mit registrierten Merkmalen des Reisepaß-Besitzers (Augenfarbe, Nasenform, Narben, Muttermale etc.) erscheint.

Handelt es sich bei diesem großen Datenbank-Fachmann gleichzeitig um einen Experten für Bildverarbeitung, für Computergrafik, für beide, oder weder für die eine, noch für die andere?

Bitte begründen Sie kurz Ihre Antwort!

Das ist ein Fachmann weder für Bildverarbeitung (weil er Bilder nicht weiter behandelt), noch für Computergrafik (weil er keinerlei synthetische Bilder generiert).

- b) Sie erhalten einen Lehrauftrag in „Grundlagen der Informatik“. Darin wollen Sie anhand der fachlichen Inhalte von Bildverarbeitung und Computergrafik den Unterschied zwischen Top-Down- und Bottom-Up-Prozessen erklären.

Welche der beiden Techniken ordnen Sie welcher der beiden Arten von Prozessen zu? (Nennungen genügen.)

Bildverarbeitung: Bottom-Up-Prozeß

Computergrafik: Top-Down-Prozeß

Zur Anschauung wollen Sie betonen, daß es vom Prinzip her verständlich wäre, wenn sich Grafik-Dateien mit Objekten oder Szenen einerseits und Bild-Dateien mit gleich aussehenden fotografischen Aufnahmen andererseits in ihrem Datenvolumen stark unterscheiden würden.

Welcher der beiden Techniken würden Sie den größeren Daten-Umfang zuordnen, und warum? (Kurze Begründung!)

Größere Dateien für Computergrafik-Objekte; denn sie enthalten (im Idealfall) die vollständige Beschreibung von Objekten / Szenen, während Bilder nur eine Ansicht gespeichert halten.

- c) Bei einer Linienziehung in einer Großanwendung mit Ihrer persönlichen Bresenham-Implementierung fällt auf, daß eine gerade Linie vom Punkt mit den Koordinaten $(x= -312; y= +501)$ zu dem Punkt bei $(x= +31; y= +723)$ fehlerhaft erscheint. Welchen Oktanten sollten Sie in Ihrer Implementierung überprüfen?

(Nennung des Oktanten und kurze Koordinatenrechnung genügen.)

Den 1. Oktanten:

$$\Delta x = (31 - (-312)) = 343;$$

$$\Delta y = (723 - 501) = 222; \Rightarrow \Delta x > \Delta y > 0.$$

- d) Nach Ihrem Erfolg in dieser Klausur bittet Sie der Leiter eines ortsnahen Planetariums um Rat:

Ein neuer Beamer soll bestellt werden. Mit ihm soll die elliptische Umlaufbahn der Erde dargestellt werden. Damit auch entfernte Sterne im Hintergrund mindestens als einzelne Pixel gezeigt werden können, muß die große Achse der Erdbahn mit mindestens 1.800 Pixeln darstellbar sein.

Wie viele Pixel (Breite x Höhe) muß der neue Beamer mindestens darstellen können, damit er den Erdumlauf aus allen Richtungen darstellen kann, ohne die vorgenannten Vorgaben zu vernachlässigen?

(Angabe von Breite x Höhe ohne Begründung genügt.)

1.800 x 1.800

- e) Woran erkennen Sie, daß die Matrix

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

keine Rotationsmatrix für die Drehung um eine der Hauptachsen (x, y, z) ist?

An der Belegung der letzten (rechten) Spalte: Sie beinhaltet auch eine Translation (vor oder nach der Rotation).

2. Aufgabe (50 Punkte + 5 Sonderpunkte)

Ein Hersteller traditioneller Jukeboxes (Abb. 2.1) will die alten Motoren durch digital gesteuerte Schrittmotoren ersetzen und fragt Sie, wieviel Rechenleistung dazu benötigt werde.



Abb. 2.1

Zur Beantwortung rechnen Sie ihm vor, wie hierzu eine Schallplatte (rechnerisch) aus ihrer Anfangsposition in der y-z-Ebene (Abb. 2.2) mit ihrem Mittelpunkt am Ende eines Hebels der Länge L, bei (y=L; z=0), an den Koordinaten-Ursprung verschoben wird (Transformationsmatrix $\mathbf{T}_{(i)}$, Abb. 2.3).

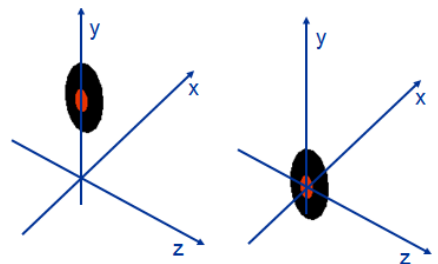


Abb. 2.2

Abb. 2.3

Anschließend wird die Schallplatte erst um $|\alpha|=90^\circ$ um die y-Achse (Transformationsmatrix $\mathbf{T}_{(ii)}$, Abb. 2.4) gedreht, so, daß ihr vorderster Punkt auf der positiven z-Achse an die positive x-Achse kommt.

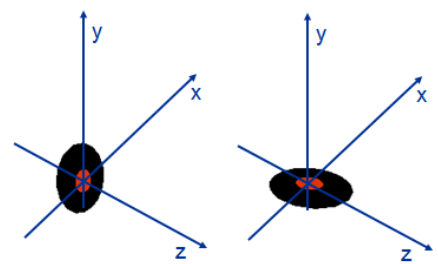


Abb. 2.4

Abb. 2.5

Danach wird sie um $|\beta|=90^\circ$ um die x-Achse (Transformationsmatrix $\mathbf{T}_{(iii)}$, Abb. 2.5) gedreht, so, daß ihr oberster Punkt auf der y-Achse an die positive z-Achse gelangt und die

Schallplatte in der (x-z-) Ebene des Plattentellers liegt.

Schließlich wird die Schallplatte entlang der z-Achse um L (die Länge des Hebels) an den Mittelpunkt des Plattentellers verschoben (Transformationsmatrix $\mathbf{T}_{(iv)}$, Abb. 2.6).

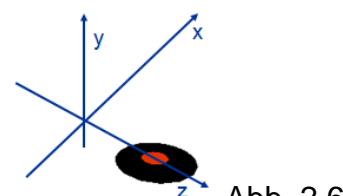


Abb. 2.6

Die Einfachheit vorstehender Überlegungen demonstrieren Sie, indem Sie die Rechenschritte der beschriebenen Animation bis zur Erlangung der Zielkoordinaten vorführen:

- a) Wie lautet die Translationsmatrix $\underline{\mathbf{T}}_{(i)}$, mit welcher die Schallplatte aus ihrer Anfangsposition mit ihrem Mittelpunkt am Ende des Tonarmes an den Koordinaten-Ursprung verschoben wird?

$$\underline{\mathbf{T}}_{(i)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -L \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- b) Geben Sie bitte (in Grad, unter Berücksichtigung des Drehsinns) den Winkel α an, um welchen die Schallplatte um die y-Achse gedreht wird, so, daß ihr vorderster Punkt von der positiven z-Achse an die positive x-Achse kommt.
Wie groß sind dann **sin α** und **cos α** ?

$$\alpha = +90^\circ \qquad \sin \alpha = 1 \qquad \cos \alpha = 0$$

- c) Wie lautet die dazugehörige Transformationsmatrix $\underline{\mathbf{T}}_{(ii)}$, die die Schallplatte um die y-Achse dreht? Geben Sie sie bitte sowohl in symbolischer (sin α , ...) als auch in arithmetischer Form an!

$$\underline{\mathbf{T}}_{(ii)} = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- d) Geben Sie nun bitte (in Grad, unter Berücksichtigung des Drehsinns) den Winkel β an, um welchen die Schallplatte um die x-Achse gedreht wird, so, daß ihr oberster Punkt auf der y-Achse an die positive z-Achse gelangt und die Schallplatte in der (x-z-) Ebene des Plattentellers liegt.
Beziffern Sie bitte auch **sin β** und **cos β** :

$$\beta = +90^\circ \qquad \sin \beta = 1 \qquad \cos \beta = 0$$

- e) Welche Transformationsmatrix $\underline{\mathbf{I}}_{(iii)}$ dreht die Schallplatte, wie oben beschrieben, um die x-Achse um den Winkel β , bis diese in der (x-z-) Ebene des Plattentellers liegt? Geben Sie sie bitte sowohl in symbolischer als auch in arithmetischer Form an!

$$\underline{\mathbf{I}}_{(iii)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- f) Wie lautet die Translationsmatrix $\underline{\mathbf{I}}_{(iv)}$, mit welcher die Schallplatte entlang der z-Achse um die Länge des Tonarms L an das Zentrum des Plattentellers verschoben wird?

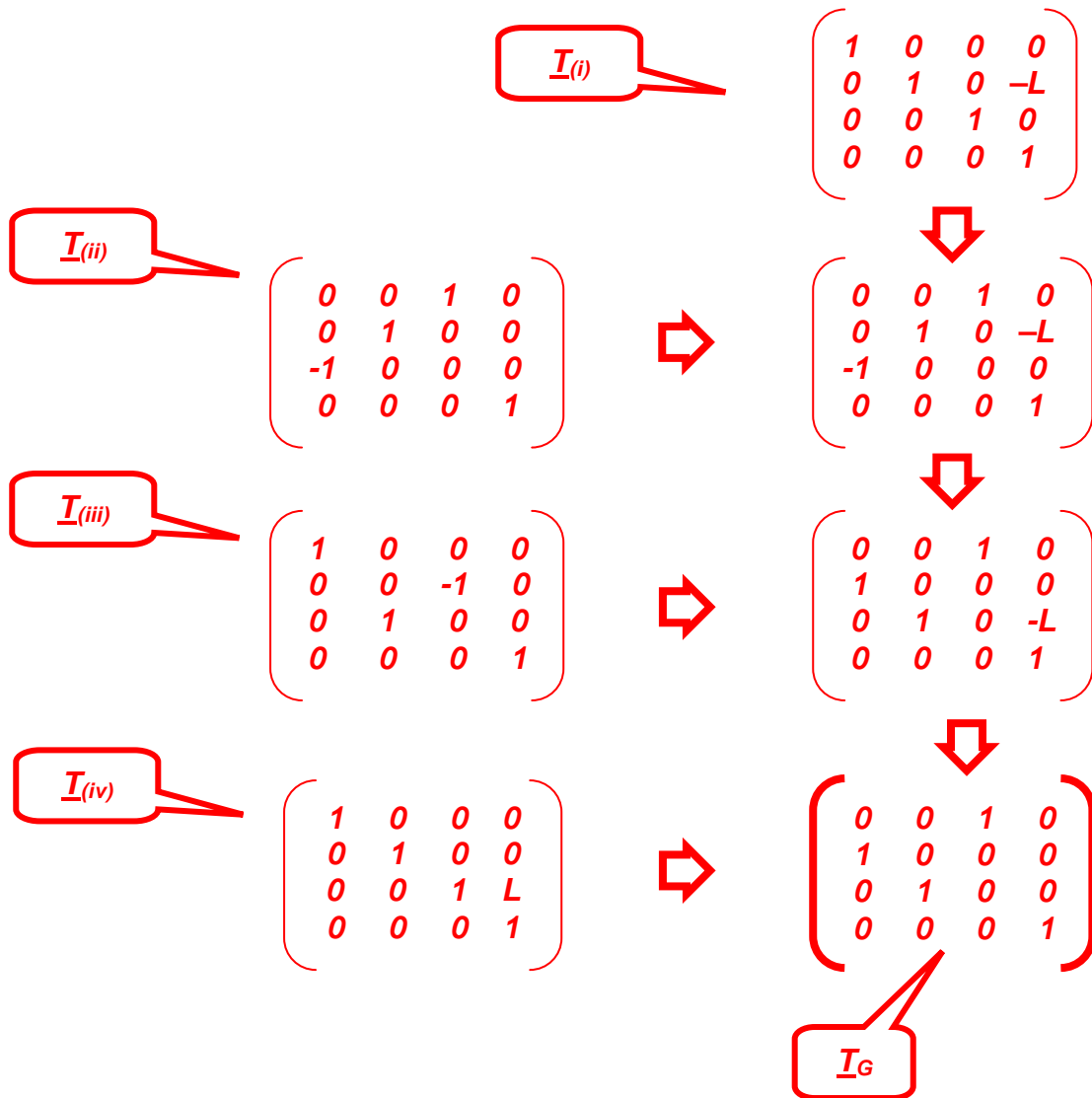
$$\underline{\mathbf{I}}_{(iv)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- g) Wie berechnet sich nun die Gesamt-Transformationsmatrix $\underline{\mathbf{I}}_{\mathbf{G}}$ aus den vorausgegangenen Transformationen?

$$\underline{\mathbf{I}}_{\mathbf{G}} = \underline{\mathbf{I}}_{(iv)} \cdot \underline{\mathbf{I}}_{(iii)} \cdot \underline{\mathbf{I}}_{(ii)} \cdot \underline{\mathbf{I}}_{(i)}$$

- h) Berechnen Sie nun bitte die Gesamt- Transformationsmatrix $\underline{\mathbf{I}}_{\mathbf{G}}$ anhand der bisher gemachten Angaben.

(b.w.)

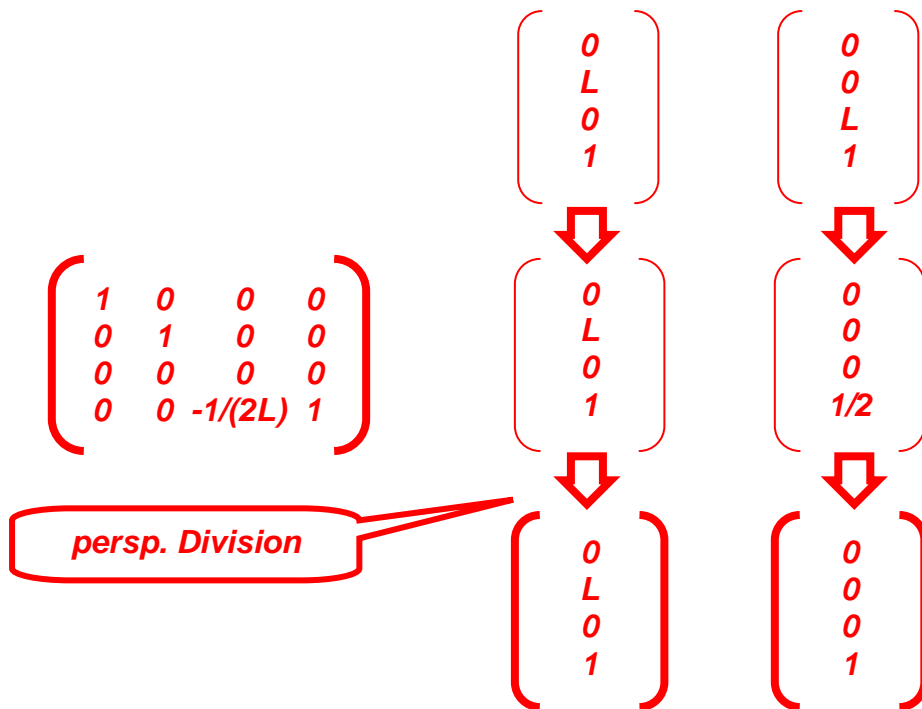


i) Weisen Sie nun bitte die Richtigkeit Ihres Vorgehens nach, indem Sie die Raumkoordinaten des Schallplatten-Mittelpunkts nach den o.a. Transformationen berechnen.

$$\begin{pmatrix} 0 \\ L \\ 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ L \\ 1 \end{pmatrix}$$

- j) Auf Nachfrage rechnen Sie vor, auf welchen Bildschirmkoordinaten der Schallplatten-Mittelpunkt abgebildet wird, wenn dieser vor und nach der o.a. Transformationsreihe (Anfangs- und Endposition) perspektivisch, mit dem Projektionszentrum bei $z=2L$, auf die x - y -Ebene projiziert wird:



Sie antworten: Vor der Animation wird der Schallplatten-Mittelpunkt auf den Bildschirm-Koordinaten:

$x = \underline{\quad 0 \quad}$, $y = \underline{\quad L \quad}$,

nach der Animation bei

$x = \underline{\quad 0 \quad}$, $y = \underline{\quad 0 \quad}$

abgebildet.

- k) (5 Sonderpunkte)

Jemand fragt Sie, wie die Auswirkung der Homogenisierung (perspektivischen Division) auf das Ergebnis anschaulich zu erklären ist.

Sie brauchen ihm nur kurze Hinweise zu geben:

Vor der Animation lag der Mittelpunkt in der Projektionsebene ($z=0$); er war zugleich seine eigene Projektion und wurde 1:1 bei $(0; L)$ abgebildet.

Nach der Animation lag derselbe Punkt auf der Sehachse, die das Projektionszentrum enthält. Er besaß somit dieselbe Projektion wie das Projektionszentrum: $(0; 0)$.

3. Aufgabe (35 Punkte + 5 Sonderpunkte)

Sie haben mit OpenGL eine virtuelle Jukebox erstellt (Abb. 3.1) und besprechen das Programm `JukeBox.c` mit einem Interessenten, der wenig von Computergrafik versteht. Sie beziehen sich dabei auf den ausgedruckten C-Code (s. Ende dieser Aufgabe).

Zunächst erklären Sie, daß die Funktion `main()` (am Code-Ende) alle organisatorischen Vorkehrungen erledigt und dazu auch `init()` aufruft, bevor die Kontrolle an GLUT übergeben wird.

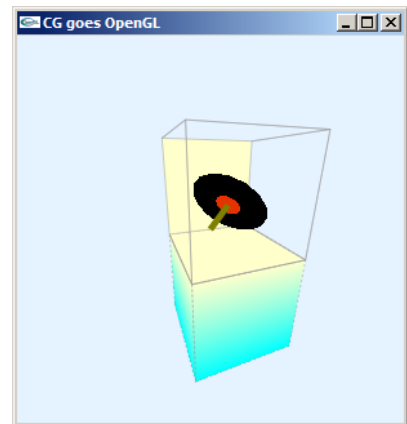


Abb. 3.1

a) Können Sie anhand des Codes der beiden Funktionen `main()` und `init()` erkennen,

- wie groß das eröffnete Fenster ist (Breite x Höhe),
- ob das Programm Double Buffering verwendet und
- ob Anti-Aliasing eingesetzt wird?

Erklären Sie bitte Ihre Antworten stichwortartig!

- **Das Fenster ist 300x300 Pixel groß (das ist die Voreinstellung, im Code ist keine Änderung erkennbar)**
- **Double Buffering wird nicht genutzt: Es müßte explizit mit `glutInitDisplayMode (GLUT_DOUBLE);` vereinbart werden, Voreinstellung ist Single Buffering.**
- **(Linien-)Anti-Aliasing wird eingesetzt – zu erkennen am Aufruf `glEnable (GL_LINE_SMOOTH);`**

b) Die Funktion `Arm()` (am Anfang der Datei) zeichnet den Hebel, der die Schallplatte festhält. Als was für ein Grafik-Primitiv ist er realisiert (Dreieck, Strich, Rechteck, Trapez, Quader, Zylinder, Kegel, Ring)?

Woran erkennen Sie das?

Als Strich – zu erkennen an der Anweisung `glBegin(GL_LINES);`

- c) Die Funktion `Record()` zeichnet die Schallplatte, die aktuell aufgelegt werden soll. Nacheinander wird hier das aufgeklebte Schild und die darunter liegende Vinyl-Platte dargestellt.

Als was für ein Grafik-Primitiv ist das Schild auf der Schallplatte realisiert (Dreieck, Strich, Rechteck, Trapez, Quader, Zylinder, Kegel, Ring)?

Woran erkennen Sie das? (Stichwortartige Erklärung genügt.)

Als Rechteck – zu erkennen an der Anweisung `glBegin(GL_POLYGON);` und den x-/y-Koordinaten (+/- LAB)

Was soll der Aufruf `glDisable(GL_CULL_FACE);` in dieser Funktion bewirken?

Dadurch ist das Plattenschild auf beiden Seiten sichtbar, ohne daß zwei Flächen definiert werden.

Als was für ein Grafik-Primitiv ist die Vinyl-Platte realisiert (Dreieck, Strich, Rechteck, Trapez, Quader, Zylinder, Kegel, Ring)? Woran erkennen Sie das?

Als Ring (Torus) – zu erkennen an der Anweisung `glutSolidTorus(LAB, REC, 2, 16);`

Wozu steht die Anweisung `glScalef()` in der Funktion `Record()`? (Kurze Erklärung)

Dadurch wird der abgebildete Reifen (Torus) in z-Richtung mit Faktor 0,1 „platt“ skaliert.

- d) Die Funktion `Box()` zieht offenbar die Kanten der Jukebox mit gestrichelten Linien nach. Erklären Sie bitte anhand der getätigten Aufrufe (und des darin enthaltenen Musters und Streckungsfaktors), wie ab Beginn jeder Linie die Pixel gesetzt bzw. ausgelassen werden.

Faktor 2, Muster 5555 (binär 0101 0101 0101 0101) bedeutet (von re. nach li. verarbeitet):

(2x1=) 2 Pixel setzen / (2x1=) 2 Pixel auslassen / ...

- e) Das Oberteil der Jukebox soll (mit Ausnahme der Rückwand) gläsern sein. Wie simuliert das Programm Glas, und woran ist das zu erkennen?

Das Oberteil wird als Drahtmodell gezeichnet – zu erkennen u.a. am Aufruf: `glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);`

- f) Welchen Index hat die Rückwand des Oberteils, die als massiv gezeichnet ist, und woran in `Box()` erkennt man das? (Stichwort genügt.)

Die Rückwand hat den Index 0 – zu erkennen an der Abfrage `if (!j1)`

- g) Das Programm `JukeBox.c` verwendet unterschiedliche Farben. Ihre Komponenten und die Funktionen, in denen sie eingestellt werden, sind unten aufgeführt. Kreuzen Sie bitte die Beschreibung an, die jeweils am ehesten zu den jeweiligen Farbtönen paßt:

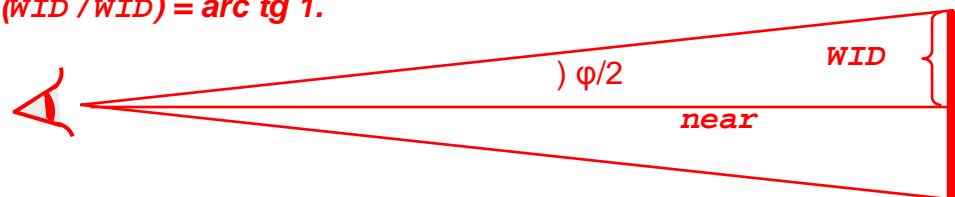
Farb-anwendung	Tonarm	Platten-schild	Nachgezo-gene Kanten	Plattenteller und Rückwand	Seiten- und Grundfläche
Funktion()	Arm()	Record()	Box()	Box()	Box()
Farbkomponen-ten (R,G,B)	.5, .5, 0.	.9, .2, .0	.6, .6, .6	1., 1., .8	0., 1., 1.
Rot, wie Tomatenmark		X			
Türkis					X
Grelles Weiß					
Tiefes Braun					
Leichtes Blau, fast Weiß					
Creme, Mattgelb				X	
Grün, „weißlich“ (ungesättigt)					
Tiefes Schwarz					
Leichtes Grau			X		
„schmutziges Gelb“, oliv	X				

h) (5 Sonderpunkte)

In der Funktion `draw()` wird das Sichtfeld vereinbart, das offenbar einem stärkeren Weitwinkel-Objektiv gleichkommt. Wie groß ist der Sichtwinkel φ in x- bzw. y-Richtung?

glFrustum (-WID, WID, -WID, WID, nah, fern); vereinbart ein Sichtfeld jeweils von -WID bis WID gleichermaßen in x- und in y-Richtung. Die Breite bzw. Höhe des Sichtfeldes gilt in Entfernung near (hier: WID) ab dem Augenpunkt; d.h., für den halben Sichtwinkel gilt:

$$\varphi/2 = \arctan(WID / WID) = \arctan 1.$$



Daraus folgt: $\varphi/2 = \pi/4 = 45^\circ$ oder $\varphi = 90^\circ$.

(Als Weitwinkel gelten Objektive mit $\varphi \geq 75^\circ$.)

```

/* JukeBox.c */
/*Darstellung eines JukeBox-Plattenwechslers mit OpenGL*/
#include <stdio.h> //wg. printf()
#include <GL/glut.h>

#define CON_CLS  "cls"
#define ESC      27
#define CR       13
#define LEN      5
#define REC      2
#define LAB      1
#define WID      (LEN+REC)

enum {X=0, Y=1, Z=2, W=3};

/*Globale Variablen: */
GLint   face[6][4]={{5,4,7,6}, {0,1,2,3}, {1,0,4,5},
                  {1,5,6,2}, {4,0,3,7}, {3,2,6,7}};
float   boxColor[6][3]={{0.f,1.f,1.}, {1.f,0.f,0.}, {0.f,1.f,1.},
                      {0.f,1.f,1.f}, {1.f,1.f,.8f}, {0.f,1.f,1.f}},
cube[8][3];
float   animAngle=0.f, delta=5.f, angle[3]={0.f,0.f,0.};
GLdouble nah, fern;

/*****
void Arm(void)
*****/
/*Tonarm am Koordinaten-Ursprung:*/
{ glLineWidth(5.);
  glColor3f(0.5f, 0.5f, 0.f);
  glBegin(GL_LINES);
    glVertex3f(0.f, 0.f, 0.);
    glVertex3f(0.f, LEN, 0.);
  glEnd();
}

/*****
void Record(void)
*****/
/*Schallplatte um Koordinaten-Ursprung:*/
{ /*Benoetigter Zeichenmodus:*/
  glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);

  /*Schild der Schallplatte:*/
  glColor3f(.9f, .2f, .0f);
  glDisable(GL_CULL_FACE);
  glBegin(GL_POLYGON);
    glVertex2f(-LAB, LAB);
    glVertex2f( LAB, LAB);
    glVertex2f( LAB, -LAB);
    glVertex2f(-LAB, -LAB);
  glEnd();
  glEnable(GL_CULL_FACE);

  /*Vinyl um z-Achse, mit Innen-/Aussenradius, nSegmenten, nSektoren:*/
  glColor3f(.0, .0, .0);
  glScalef(1.f, 1.f, .1f);
  glutSolidTorus(LAB, REC, 2, 16);
}

```

```

/*****/
void Box(void)
/*****/
/*Box aus zwei Einheitswuerfeln mit vorgegebenen Farben zeichnen:*/
{ int j1 = 0, j2=0, jj=0, count=0;

/*Benoetigte Linienstaecke:*/
glLineWidth(.5);

/*Unterteil 2x zeichnen*/
for (jj = 0; jj < 2; jj++)
{ if (!jj) { glPolygonMode(GL_FRONT_AND_BACK, GL_FILL); }
  else    { glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
            glEnable(GL_LINE_STIPPLE);
            glLineStipple(2, 0x5555);
            glColor3f(.6f, .6f, .6f);
          }

/*Alle Flaechen des Unterteils:*/
for (j1 = 0; j1 < 6; j1++)
{ /*Alle Eckpunkte der Flaechen:*/
  glBegin(GL_POLYGON);
  for (j2 = 0; j2 < 4; j2++)
  { if (!jj)
    { if (cube[face[j1]][j2]][Y] > 0.f)
      glColor3f(1.f, 1.f, .8f);
      else glColor3f(0.f, 1.f, 1.f);
    }
    glVertex3fv(cube[face[j1]][j2]);
  }
  glEnd();
}
}
glDisable(GL_LINE_STIPPLE);

/*Oberteil zeichnen, verschoben zum Unterteil:*/
glTranslatef(0.f, 1.f, 0.f);

/*Alle Flaechen des Oberteils:*/
glDisable(GL_CULL_FACE);
for (j1 = 0; j1 < 6; j1++)
{ if (!j1) { glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
            glColor3f(1.f, 1.f, .8f); }
  else    { glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
            glColor3f(.6f, .6f, .6f); }

/*Alle Eckpunkte der Flaechen:*/
glBegin(GL_POLYGON);
for (j2 = 0; j2 < 4; j2++)
{ glVertex3fv(cube[face[j1]][j2]);
}
glEnd();
}
glEnable(GL_CULL_FACE);
}

/*****/
void draw(void)
/*****/
{ float edgeLen= LEN + LAB + REC;
  nah = WID; fern = nah + 3*(LEN + LAB+REC);
}

```

```

/*Sichtfeld:*/
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glFrustum (-WID, WID, -WID, WID, nah, fern);

/*Fenster loeschen:*/
glClear(GL_COLOR_BUFFER_BIT);

/*Animation, Skalierung neu aufbauen (ohne Fehlerfortpflanzung):*/
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

/*Alles ins Sichtvolumen verschieben:*/
glTranslatef(0.f, 0.f, -(nah+fern)/2.);

/*Positionierung JukeBox:*/
glRotatef(angle[X], 1.f, 0.f, 0.f);
glRotatef(angle[Y], 0.f, 1.f, 0.f);
glRotatef(angle[Z], 0.f, 0.f, 1.f);

/*JukeBox skalieren & verschieben, wieder vergessen:*/
glPushMatrix();
/*Mitte der Rueckseite an den Koord.-Ursprung:*/
glTranslatef(0.f, -edgeLen/2., edgeLen/2.);
glScalef(edgeLen, edgeLen, edgeLen);
Box();
glPopMatrix();

/*Platte an Tonarm setzen & animieren, beide an Box animieren,
wieder vergessen:*/
glRotatef(animAngle, 1.f, 0.f, 0.);

/*Positionierung Schallplatte an Tonarm:*/
glPushMatrix();
glRotatef(animAngle, 0.f, 1.f, 0.);
glTranslatef(0.f, LEN, 0.);
glRotatef(90.f, 0.f, 1.f, 0.);
Record();
glPopMatrix();
Arm();

/*Bild fertig:*/
glFlush();
}

/*****
void key(unsigned char key, int x, int y)
*****/
/*Menue und Eingabe-Behandlung:*/
{ switch (key)
  { case ESC: exit(0);
    case CR : animAngle += delta;
              if (delta > 0. && animAngle >= 90.
                  || delta < 0. && animAngle <= 0.) delta = -delta; break;
    case 'x': angle[X] -= 5.; if (angle[X] <=-360) angle[X]+=360; break;
    case 'X': angle[X] += 5.; if (angle[X] >= 360) angle[X]-=360; break;
    case 'y': angle[Y] -= 5.; if (angle[Y] <=-360) angle[Y]+=360; break;
    case 'Y': angle[Y] += 5.; if (angle[Y] >= 360) angle[Y]-=360; break;
    case 'z': angle[Z] -= 5.; if (angle[Z] <=-360) angle[Z]+=360; break;
    case 'Z': angle[Z] += 5.; if (angle[Z] >= 360) angle[Z]-=360; break;
  }
}

```

```
system (CON_CLS);
printf ("\n\r Press (keeping the GLUT window activated):");
printf ("\n\n\r <ESC>  to quit");
printf ("\n\r <CR>  to change the record");
printf ("\n\r x / X  decrease/increase X-rotation angle");
printf ("\n\r y / Y  decrease/increase Y-rotation angle");
printf ("\n\r z / Z  decrease/increase Z-rotation angle");
printf ("\n\n\r Current values:");
printf ("\n\r          angle[X]=%7.2f", angle[X]);
printf ("\n\r          angle[Y]=%7.2f", angle[Y]);
printf ("\n\r          angle[Z]=%7.2f", angle[Z]);
draw();
}

/*****
void init(void)
*****/
/*Einheitswürfel:*/
{ cube[0][X] = cube[3][X] = cube[4][X] = cube[7][X] = -.5;
  cube[1][X] = cube[2][X] = cube[5][X] = cube[6][X] = .5;
  cube[0][Y] = cube[1][Y] = cube[4][Y] = cube[5][Y] = -.5;
  cube[2][Y] = cube[3][Y] = cube[6][Y] = cube[7][Y] = .5;
  cube[0][Z] = cube[1][Z] = cube[2][Z] = cube[3][Z] = .5;
  cube[4][Z] = cube[5][Z] = cube[6][Z] = cube[7][Z] = -.5;

  /*Loeschfarbe setzen:*/
  glClearColor(.9f, .95f, 1.f, 1.f);

  /*Schattierung, Linienziehung als Programm-Standards:*/
  glShadeModel(GL_SMOOTH);
  glEnable(GL_LINE_SMOOTH);
  glEnable(GL_BLEND);
  glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

  /*Tastendruck simulieren, um Menue auszugeben:*/
  key(' ', 0, 0);
}

/*****
int main(int argc, char **argv)
*****/
{ glutInit(&argc, argv);
  glutCreateWindow("CG goes OpenGL");
  glutDisplayFunc(draw);
  glutKeyboardFunc(key);
  init();
  glutMainLoop();
  return 0;
}
```