

Computergrafik I

WS 2003/04

Dokumentation des OpenGL Projekts

Holger Gerth
Technische Universität Chemnitz
holger.gerth@informatik.tu-chemnitz.de

0. Inhalt

1. Beschreibung des Modells
2. Beschreibung der Koordinatensysteme / Transformationen
3. Szenegraph
4. Material- und Lichtdefinitionen
5. Einstellungen
6. Interaktion mit der Szene
7. Texturen
8. sonstige Informationen
9. Screenshots des Modells

1. Beschreibung des Modells

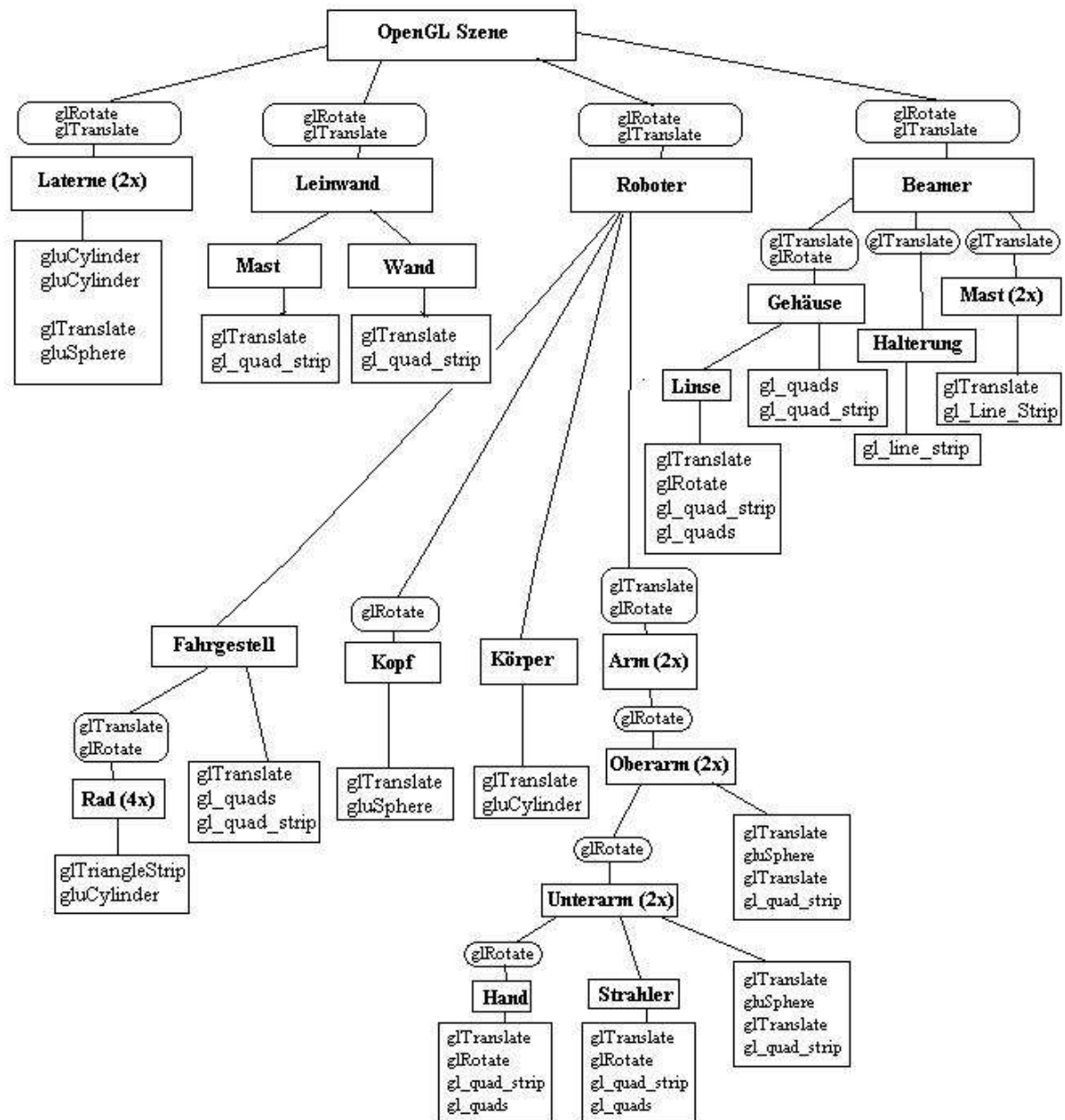
Die Szene stellt einen Roboter dar, der sich auf einer Plattform bewegen kann. Zusätzlich enthält die Szene noch einen Projektor (Beamer) samt Leinwand und Halterung sowie zwei Laternen.

2. Beschreibung der Koordinatensysteme / Transformationen

- Ursprungskoordinatensystem (Kamera-Viewpoint)
(0.0,0.0,0.0) --> zu Beginn
- Koordinatensystem Roboter
(10.0,0.0,10.0) --> zu Beginn
- Koordinatensystem Laterne 1
(-10.0,0.0,-5.0)
- Koordinatensystem Laterne 2
(10.0,0.0,-5.0);
- Koordinatensystem Beamer
(-20.0,0.0,24.0) --> Mast links
(20.0,0.0,24.0) --> Mast rechts
(0.0,11.0,24.0) --> Halterung
(0.0,8.0,24.0) --> Gehäuse mit Linse
- Koordinatensystem Leinwand
(-4.0,10.0,-23.98) -- Leinwand
(0.0,0.0,-24.5) --> Mast

Kamera-Viewpoint sowie der Standort des Roboters können durch Interaktion mit dem Modell verändert werden.

3. Szenegraph



4. Material- und Lichtdefinitionen

4.1.Lichtdefinitionen

// 1. Lichtquelle (GL_LIGHT0; Parallelstrahler)

LAmbient[4] = {0.1F,0.1F,0.1F,1.0F};

LDiffuse[4] = {0.7F,0.7F,0.7F,1.0F};

LSpecular[4] = {0.3F,0.3F,0.3F,1.0F};

LPosition[4] = {0.5F,1.0F,0.0F,0.0F};

LSpotDirec[3] = {0.0F,0.0F,0.0F};

LSpotCutOff = 180.0F;

LSpotExponent = 0.0F;

LAttenuationConst = 1.0F;

LAttenuationLinear = 0.0F;

LAttenuationQuadrat = 0.0F;

// 2. & 3. Lichtquelle (GL_LIGHT1 & GL_LIGHT2; Punktstrahler; Laternen)

LAmbient1[4] = {0.02F,0.02F,0.02F,1.0F};

LDiffuse1[4] = {1.0F,1.0F,1.0F,1.0F};

LSpecular1[4] = {0.3F,0.3F,0.3F,1.0F};

LSpotCutOff1 = 180.0F;

LSpotExponent1 = 0.0F;

LAttenuationConst1 = 0.8F;

LAttenuationLinear1 = 0.08F;

LAttenuationQuadrat1 = 0.03F;

// 1. Laterne

LPosition1[4] = {0.0F,0.0F,5.4F,1.0F};

LSpotDirec1[3] = {0.0F,0.0F,-5.4F};

// 2. Laterne

LPosition2[4] = {0.0F,0.0F,5.4F,1.0F};

LSpotDirec2[3] = {0.0F,0.0F,-5.4F};

// 4. Lichtquelle (GL_LIGHT3; Spot-Strahler; Strahler des Robo)

LAmbient3[4] = {0.0F,0.0F,0.0F,1.0F};

LDiffuse3[4] = {1.0F,0.0F,0.0F,1.0F};

LSpecular3[4] = {1.0F,0.0F,0.0F,1.0F};

LPosition3[4] = {0.0F,0.0F,0.04F,1.0F};

LSpotDirec3[3] = {0.0F,0.0F,-0.04F};

LSpotCutOff3 = m_structMoveRobo.SpotCutOff; // start bei 30.0f

LSpotExponent3 = m_structMoveRobo.SpotExponent; // start bei 5.0f

LAttenuationConst3 = 0.1F;

LAttenuationLinear3 = 0.05F;

LAttenuationQuadrat3 = 0.0F;

// 5. Lichtquelle (GL_LIGHT4; Spot-Strahler; Beamer)

LAmbient4[4] = {0.0F,0.0F,0.0F,1.0F};

LDiffuse4[4] = {0.8F,0.9F,0.8F,1.0F};

LSpecular4[4] = {0.2F,0.2F,0.2F,1.0F};

LPosition4[4] = {0.0F,0.0F,0.0F,1.0F};

LSpotDirec4[3] = {0.0F,1.0F,0.0F};

LSpotCutOff4 = 7.0F;

LSpotExponent4 = 10.0F;
 LAttenuationConst4 = 0.2F;
 LAttenuationLinear4 = 0.05F;
 LAttenuationQuadrat4 = 0.0F;

4.2. Materialdefinitionen

- Global:
 - MAmbient[4] = {0.9F,0.9F,0.9F,1.0F};
 - GLfloat MDiffuse[4] = {0.5F,0.5F,0.5F,1.0F};
 - GLfloat MSpecular[4] = {0.0F,0.0F,0.0F,1.0F};

Die Materialdefinitionen werden über die Funktion

bool COpenGL::SetMatCol(fRGBAColor *pCurMatColor, GLenum MatFace,
GLfloat Ca, GLfloat Cd, GLfloat Cs)

erstellt, indem Konstanten für diffuses, spekulares und ambientes Licht an die jeweilige Materialfarbe multipliziert werden. Dazu nun die Aufrufe der Funktion für die verschiedenen Materiale:

- Grundfläche
 - SetMatCol(&m_fStdFarben.OLIVGRUEN,GL_FRONT,0.80F,0.60F,0.10F);
- Roboter (allgemein)
 - SetMatCol(&m_fStdFarben.SILBER,GL_FRONT,0.95F,0.95F,0.40F);
- Roboterfahrgestell
 - SetMatCol(&m_fStdFarben.GRAU,GL_FRONT,0.80F,0.80F,0.40F);
- Roboterräder
 - SetMatCol(&m_fStdFarben.GRAU,GL_FRONT,0.50F,0.50F,0.50F);
 - SetMatCol(&m_fStdFarben.SILBER,GL_FRONT,0.90F,0.80F,0.20F);
 - SetMatCol(&m_fStdFarben.CYAN,GL_FRONT,0.90F,0.80F,0.20F);
 - SetMatCol(&m_fStdFarben.SCHWARZ,GL_FRONT,0.40F,0.20F,0.00F);
- Robostrahlerglas
 - SetMatCol(&m_fStdFarben.ROT,GL_FRONT,0.90F,0.90F,0.90F);
- Laternenmast & Halterung
 - SetMatCol(&m_fStdFarben.SILBER,GL_FRONT,0.90F,0.80F,0.20F);
- Laternenglas
 - SetMatCol(&m_fStdFarben.WEISS,GL_FRONT,0.90F,0.90F,0.90F);
- Leinwand
 - SetMatCol(&m_fStdFarben.GRAU,GL_FRONT,0.90F,0.90F,0.70F);
- Leinwandmast
 - SetMatCol(&m_fStdFarben.SILBER,GL_FRONT,0.90F,0.90F,0.70F);
- Beamergehäuse
 - SetMatCol(&m_fStdFarben.SILBER,GL_FRONT,0.90F,0.90F,0.70F);
- Beamerglas
 - SetMatCol(&m_fStdFarben.WEISS,GL_FRONT,0.90F,0.90F,0.90F);
- Beamerast
 - SetMatCol(&m_fStdFarben.SILBER,GL_FRONT,0.90F,0.90F,0.70F);
- Beamerhalterung
 - SetMatCol(&m_fStdFarben.BRAUN,GL_FRONT,0.90F,0.70F,0.00F);

4.3. Farbdefinitionen

Weiss	= {1.0F,1.0F,1.0F,0.7F};
Schwarz	= {0.0F,0.0F,0.0F,0.7F};
Rot	= {1.0F,0.0F,0.0F,0.7F};
Gruen	= {0.0F,1.0F,0.0F,0.7F};

Blau	= {0.0F,0.0F,1.0F,0.7F};
Gelb	= {1.0F,1.0F,0.0F,0.7F};
Cyan	= {0.0F,1.0F,1.0F,0.7F};
Magenta	= {1.0F,0.0F,1.0F,0.7F};
Orange	= {1.0F,0.5F,0.0F,0.7F};
Braun	= {0.5F,0.25F,0.0F,0.7F};
Lila	= {0.5F,0.0F,0.5F,0.7F};
Olivgruen	= {0.3F,0.5F,0.0F,0.7F};
Dunkelgruen	= {0.0F,0.5F,0.0F,0.7F};
Marineblau	= {0.0F,0.0F,0.5F,0.7F};
Tuerkis	= {0.0F,0.5F,0.5F,0.7F};
Backsteinrot	= {0.75F,0.0F,0.0F,0.7F};
Grau	= {0.5F,0.5F,0.5F,0.7F};
Silber	= {0.75F,0.75F,0.75F,0.7F};

5. Einstellungen

Sämtliche Einstellungen der OpenGL-Zustände erfolgen über die Buttons des MFC-Fenster:

- Init - Initialisiert das OpenGL-Fenster
- Lichteffekte an/aus - Schaltet GL_LIGHTING an oder ab (Voreinstellung: an)
- Laterne - Schaltet die Laternen an oder aus (Voreinstellung: aus)
- Parallelstrahler - Schaltet den globalen Parallelstrahler an oder ab (Voreinstellung: aus)
- Robo-Strahler - Schaltet den Spot-Strahler des Roboters an oder ab (Voreinstellung: aus)
- Beleuchtungsmodus - Wechselt zwischen GL_COLOR und GL_COLOR_MATERIAL Modus (Voreinstellung: GL_COLOR)
- Shading - Schaltet zwischen Flat- und Smooth-Shading um (Voreinstellung: Smooth-Shading)
- Transparenz - Schaltet die gesamte Szene transparent (Voreinstellung: aus)
- Gitternetzdarstellung - Stellt die Szene als Gitternetz dar (GL_LINE) (Voreinstellung: GL_FILL)
- Z-Buffer - Schaltet den Z-Buffer-Algorithmus an oder ab (Voreinstellung: an)
- Texturen - Schaltet die Texturierung an oder ab (Voreinstellung: an)
- Nebel - Schaltet den Nebel (glFog) an oder ab (Voreinstellung: an)
- Beamer - Schaltet das Beamerbild sowie den Spot-Strahler des Beamers an oder ab (Voreinstellung: aus)
- Koordinatensystem - Stellt das Szenenkoordinatensystem dar (Voreinstellung: aus)
- Projektionsart - Schaltet zwischen Paraller Projektion (glOrtho) und Zentraler Projektion (glFrustum) um (Voreinstellung: Zentralprojektion)

6. Interaktion

Über folgende Tasten kann eine zusätzliche Interaktion mit dem Modell vorgenommen werden.

[W]	-	Roboter nach vorwärts bewegen
[S]	-	Roboter nach rückwärts bewegen
[A]	-	Roboter nach links drehen
[D]	-	Roboter nach rechts drehen
[O]	-	Kopf nach links bewegen
[P]	-	Kopf nach rechts bewegen
[Q]	-	linker Arm hoch
[Y]	-	linker Arm runter
[E]	-	rechter Arm hoch
[X]	-	rechter Arm runter
[R]	-	rechter Unterarm hoch
[T]	-	rechter Unterarm runter
[Z]	-	mit der Hand „zugreifen“
[H]	-	Hand rechts herum drehen
[J]	-	Hand links herum drehen
[F]	-	linker Unterarm hoch (mit Strahler)
[G]	-	linker Unterarm runter (mit Strahler)
[V]	-	RoboStrahler Spot Exponent erhöhen
[B]	-	RoboStrahler Spot Exponent verringern
[N]	-	RoboStrahler CutOff erhöhen
[M]	-	RoboStrahler CutOff verringern

Die Kameratransformationen werden komplett über die Mouse gesteuert.

Mittels gedrückter linker Mousetaste und Bewegen der Mouse nach Oben oder Unten dreht sich die Kamera vertikal um die aktuelle Position. Bei Links und Rechts erfolgt die Drehung horizontal.

Mittels gedrückter rechter Mousetaste und Bewegung der Mouse nach Oben bzw. Unten bewirkt man ein Bewegen der Kamera nach Vorn bzw. nach Hinten.

Clipplanes verändern:

[1]	-	linke Clipplane vergrößern (left plane)
[2]	-	linke Clipplane verkleinern
[3]	-	rechte Clipplane vergrößern (right plane)
[4]	-	rechte Clipplane verkleinern
[5]	-	obere Clipplane vergrößern (top plane)
[6]	-	obere Clipplane verkleinern
[7]	-	untere Clipplane vergrößern (bottom plane)
[8]	-	untere Clipplane verkleinern
[9]	-	vordere Clipplane vergrößern (near plane)
[0]	-	vordere Clipplane verkleinern
[U]	-	hintere Clipplane vergrößern (far plane)
[I]	-	hintere Clipplane verkleinern

7. Texturen

Texturen werden mittels der Funktion

```
bool COpenGL::LoadPrepareCurTexture(CString Filename, GLuint *pTexName,  
                                     GLint MinFilter, GLint MagFilter)
```

geladen. Hier nun die verwendeten Texturen samt Filter:

- Grundfläche
 LoadPrepareCurTexture("Texturen/rasen.bmp",
 &m_TexturNamen.Rasen, GL_NEAREST, GL_NEAREST);
- Körper des Roboters
 LoadPrepareCurTexture("Texturen/koerper.bmp",
 &m_TexturNamen.RoboKoerper, GL_LINEAR, GL_LINEAR);
- Kopf des Roboters
 LoadPrepareCurTexture("Texturen/kopf.bmp",
 &m_TexturNamen.RoboKopf, GL_LINEAR, GL_LINEAR);
- Beamerbild
 LoadPrepareCurTexture("Texturen/beamer.bmp",
 &m_TexturNamen.BeamerBild, GL_LINEAR, GL_LINEAR_MIPMAP_NEAREST);

8. Sonstige Informationen

Die Initialisierung der OpenGL erfolgt über die Win32-API.

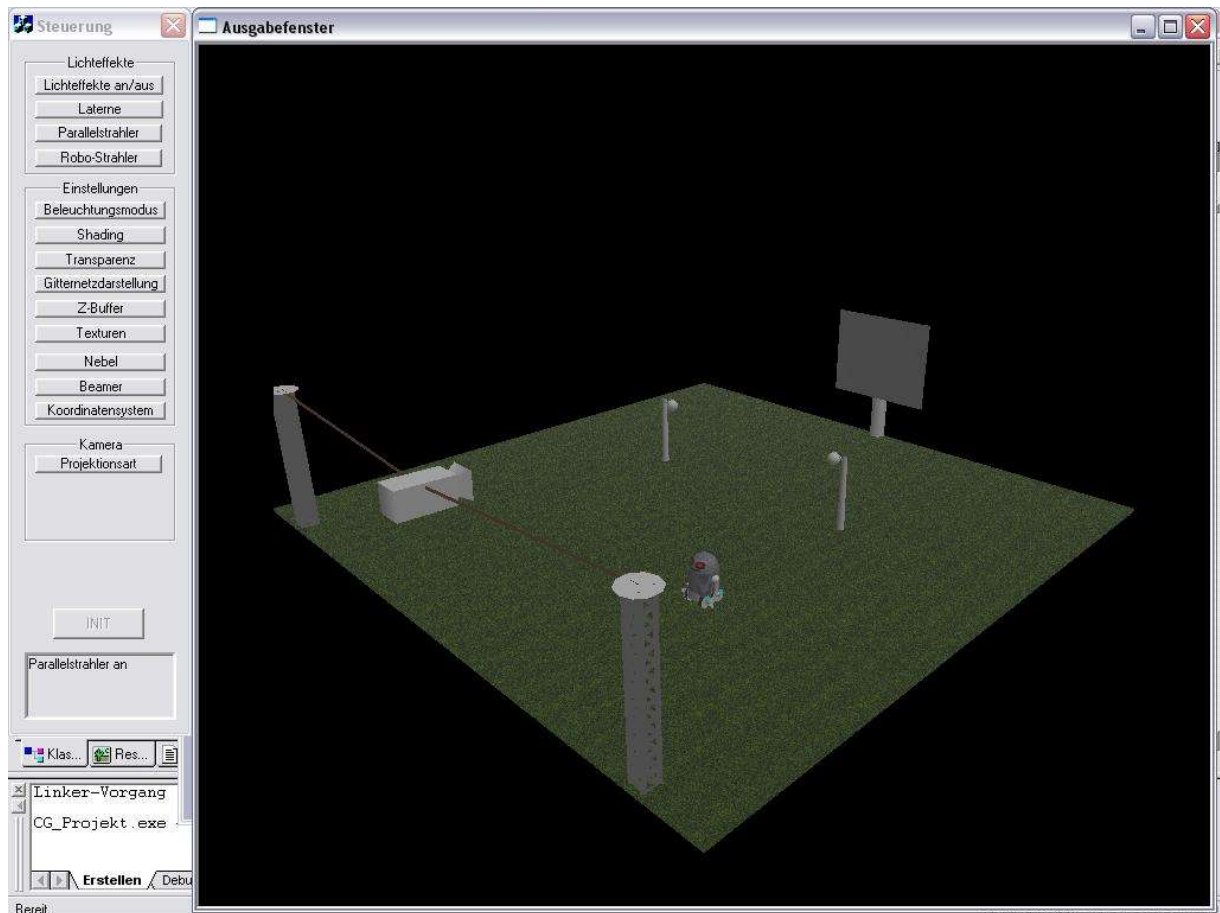
Die Interaktion mit dem Modell wurde über MFC realisiert.

Die Texturen werden im Unterverzeichnis Texturen (relativ zum Verzeichnis der Executable) erwartet.

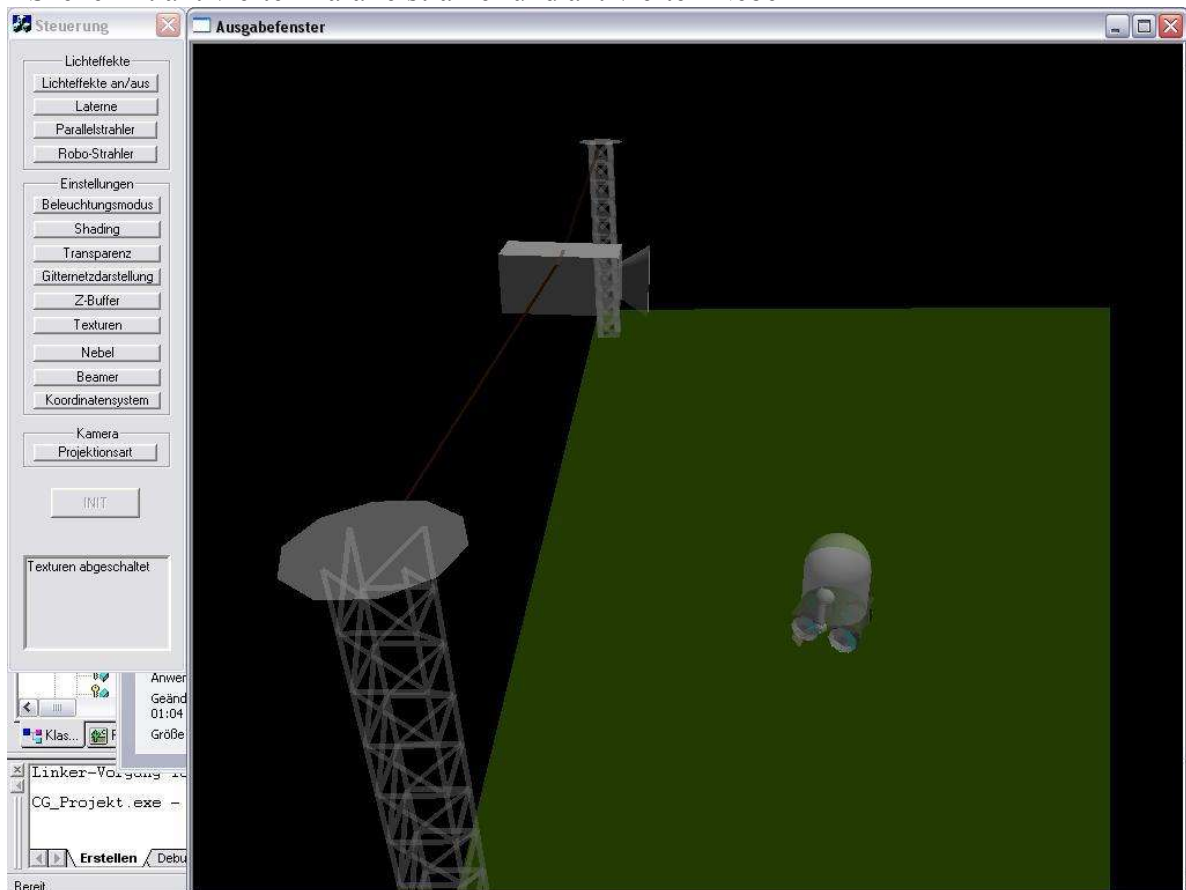
Zusätzlich zu Vorlesung und Übung wurden noch folgende Quellen verwandt:

- Lorenz Burggraf - „Jetzt lerne ich OpenGL“
- Nehe Game Tutorial - <http://www.nehe.gamedev.net>
- OpenGL FAQ - <http://www.3dsourc.de>

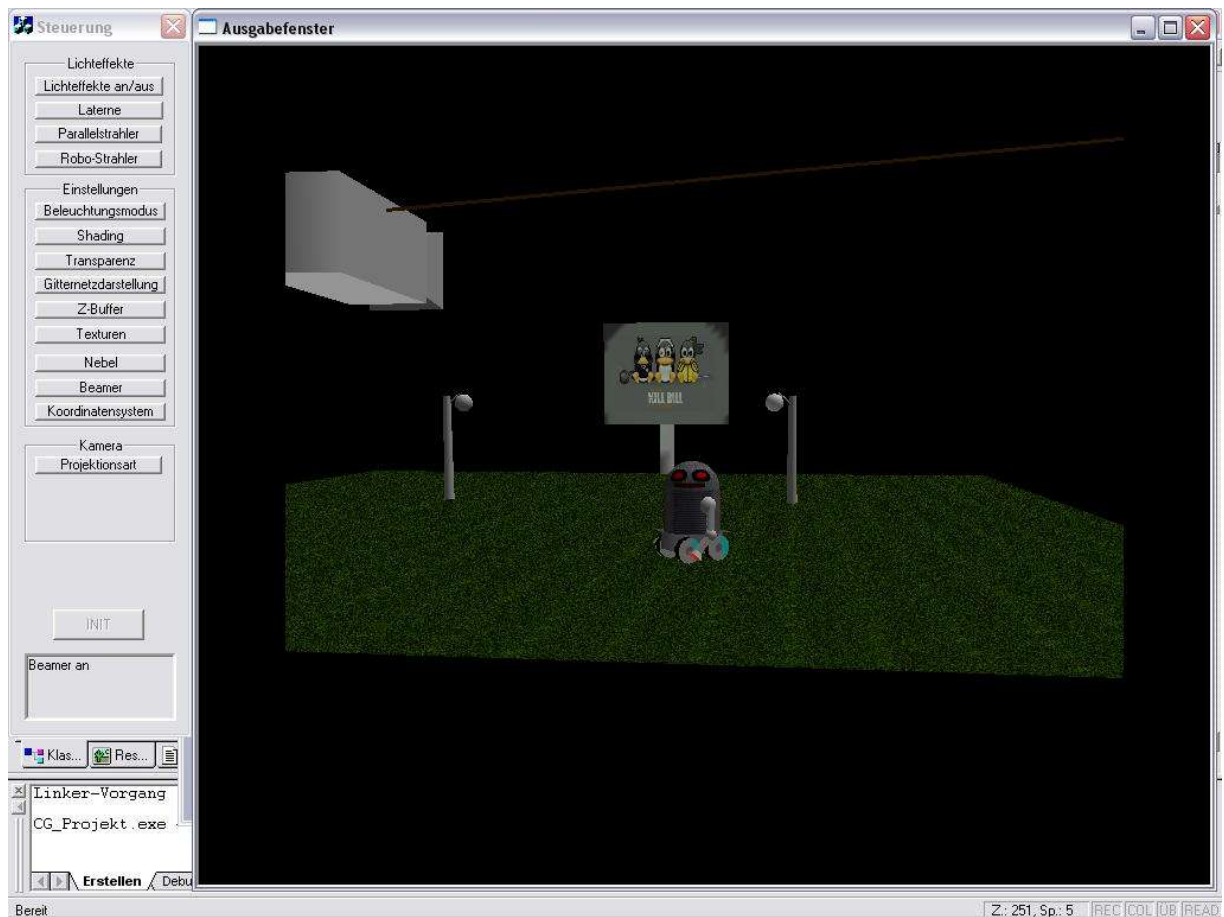
2. Screenshots



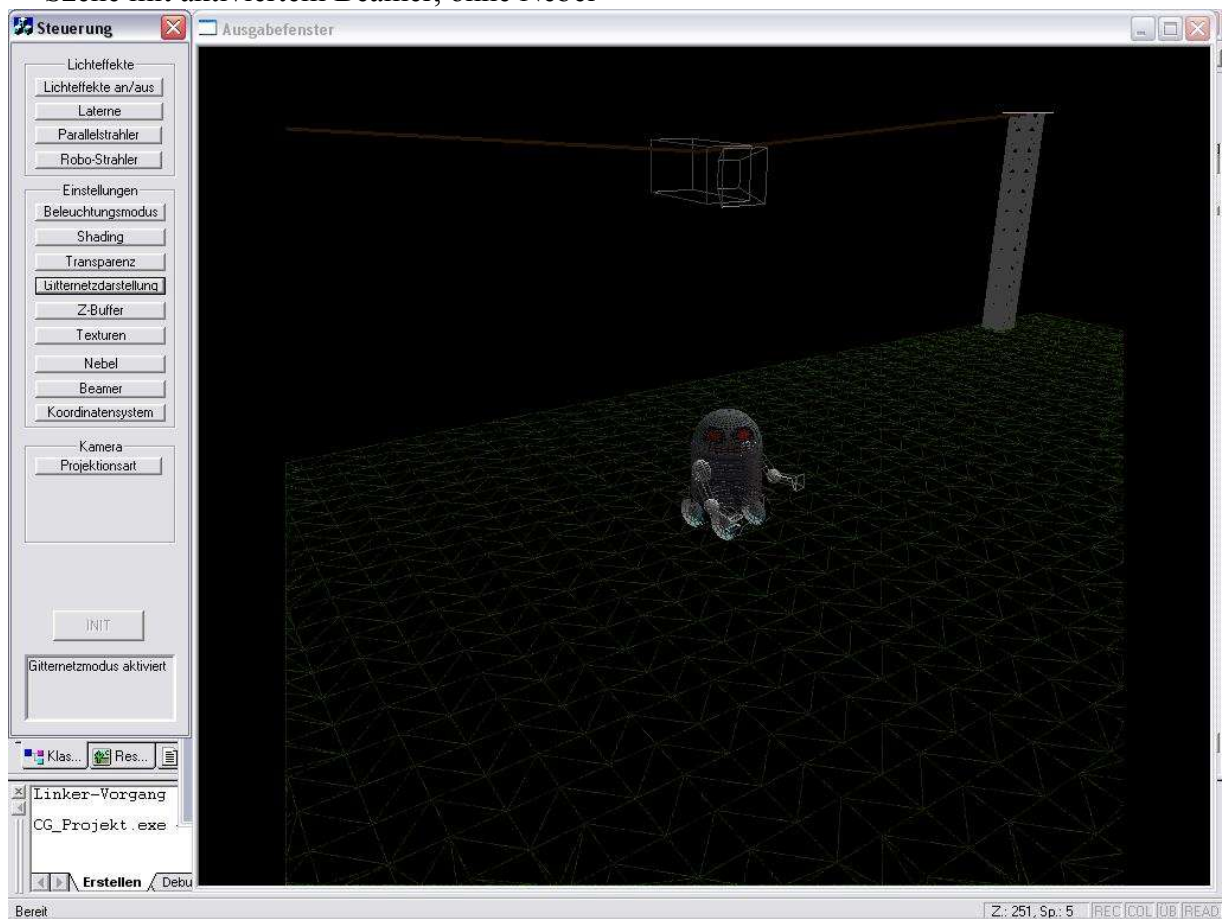
Szene mit aktivierten Parallelstrahler und aktivierten Nebel



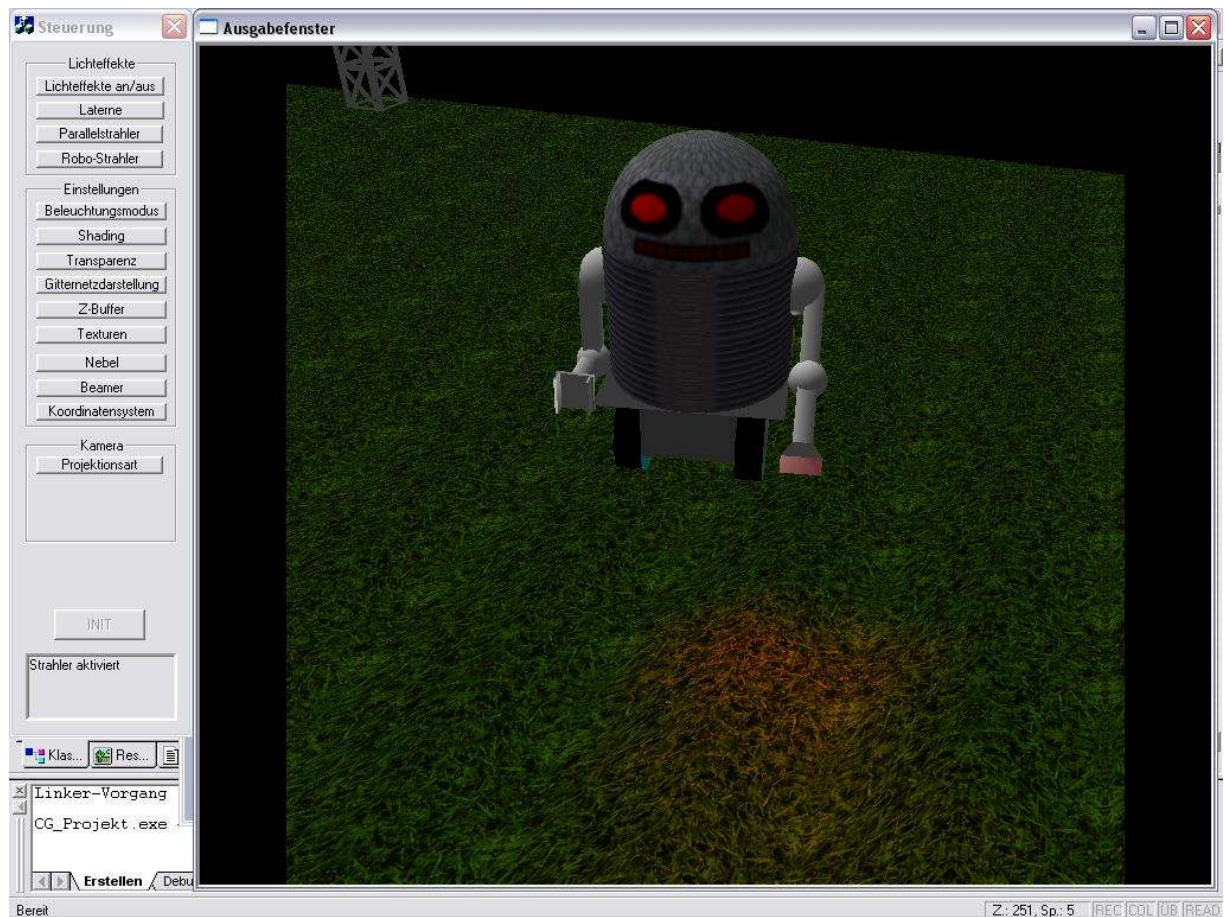
Szene transparent, Texturen und Nebel deaktiviert



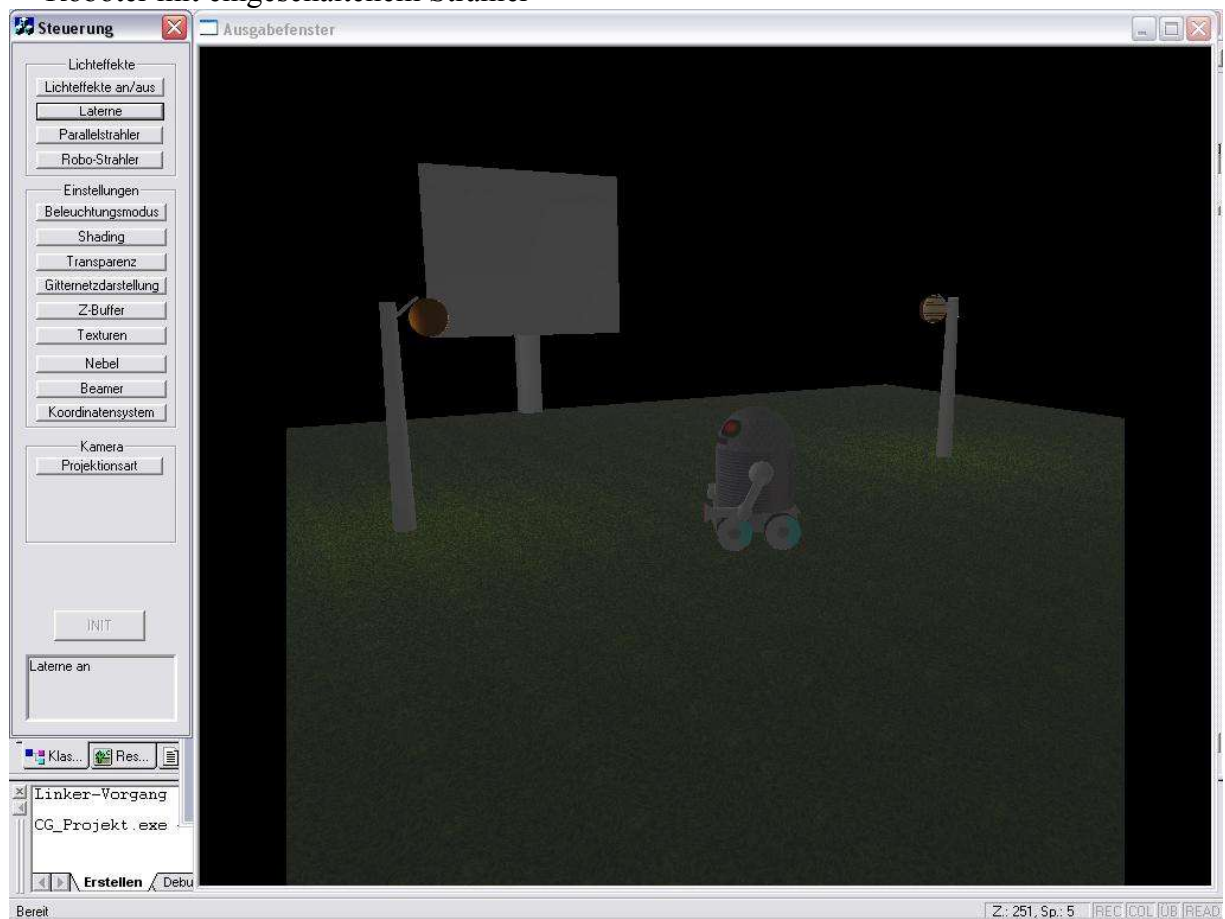
Szene mit aktiviertem Beamer, ohne Nebel



Szene in Gitternetzdarstellung



Roboter mit eingeschaltetem Strahler



Szene mit aktiviertem Nebel und aktivierten Laternen