

Übungen zu Graphikprogrammierung in C++

Abgabetermin: 14. Dezember 2004 *Was man wissen muss:*

- *Rotationen und Translationen in 3D*
- *OpenGL Modelview Matrix*

Aufgabe 18 (H) Teapot

GLUT bietet eine Funktion `glutSolidTeapot(GLfloat size)` die eine Teekanne rendert. In dieser Aufgabe soll eine Anwendung (`mainA18.cpp`) geschrieben werden, die diesen Teapot rendert.

Die `mainA15.cpp` kann hierfür als Vorlage verwendet werden. Zusätzlich zu der Teekanne soll eine Tischplatte mittels der Klasse `Box` erstellt werden. Des weiteren soll der Tisch mit Füßen bestückt werden. Als Zusatzaufgabe kann noch eine Tasse implementiert werden um das Teeset komplett zu machen. Versuchen Sie dabei die Bibliothek `3Dshape` so zu erweitern, dass moeglichst keine `glVertex3f`-Aufrufe in `mainA18.cpp` durchgeführt werden müssen.

Mittels Maus soll die komplette Szene rotiert werden. Wird die Maus bei gedrückter linker Maustaste nach oben, unten, rechts oder links bewegt, so dreht sich die ganze Szene (Teekanne inklusive dem Tisch).

Die Aufgabe soll dann noch um eine Funktion erweitert werden. Durch das Drücken der Taste 'a' soll eine Bewegung simuliert werden, mit der Tee aus der Kanne gegossen wird.

Aufgabe 19 (H) Licht

Das Model der Teekanne erscheint noch sehr künstlich. Durch das Implementieren von verschiedenen Lichtquellen wird die Kanne realistischer. Wir unterscheiden zwischen zwei Arten von Licht - ambientem und diffusem Licht.

Ambientes Licht beleuchtet die Szene von keinem definierten Punkt aus, sondern die komplette Szene homogen.

Diffuses Licht kommt von einer punktförmigen Lichtquelle. Diese Lichtquelle wird durch eine dreidimensionale Position im Raum definiert. Der folgende Code implementiert sowohl eine Lichtquelle für ambientes als auch diffuses Licht.

```
GLfloat LightAmbient[]= { 0.5f, 0.5f, 0.5f, 1.0f };  
GLfloat LightDiffuse[]= { 1.0f, 1.0f, 1.0f, 1.0f };  
GLfloat LightPosition[]= { 0.0f, 0.0f, 2.0f, 1.0f };
```

```
glLightfv(GL_LIGHT1, GL_AMBIENT, LightAmbient);  
glLightfv(GL_LIGHT1, GL_DIFFUSE, LightDiffuse);
```

```
glLightfv(GL_LIGHT1, GL_POSITION, LightPosition);
glEnable(GL_LIGHT1);
glEnable(GL_LIGHTING);
glEnable(GL_COLOR_MATERIAL);
```

In dieser Aufgabe soll nun die Teekanne aus Aufgabe 19 erweitert werden. Durch das Drücken der Taste 'l' soll das Licht gesetzt werden. Mit den Tasten '+' und '-' soll nun der Anteil des ambienten Lichtes erniedrigt bzw. erhöht werden.

In OpenGL gibt es die Möglichkeit Nebel zu erzeugen. Im folgenden wird ein kurzer Beispielcode zum Erzeugen von Nebel aufgeführt.

```
GLuint fogMode[] = { GL_EXP, GL_EXP2, GL_LINEAR };
GLuint fogfilter = 0;
GLfloat fogColor[4] = {0.5f, 0.5f, 0.5f, 1.0f};

glClearColor(0.5f, 0.5f, 0.5f, 1.0f); // We'll Clear To The Color Of The Fog ( Modified )
glFogi(GL_FOG_MODE, fogMode[fogfilter]); // Fog Mode
glFogfv(GL_FOG_COLOR, fogColor); // Set Fog Color
glFogf(GL_FOG_DENSITY, 0.25f); // How Dense Will The Fog Be
glHint(GL_FOG_HINT, GL_DONT_CARE); // Fog Hint Value
glFogf(GL_FOG_START, 5.0f); // Fog Start Depth
glFogf(GL_FOG_END, 10.f); // Fog End Depth
glEnable(GL_FOG); // Enables GL_FOG
```

Die bisher implementierte Teekannenszene soll nun um Nebel erweitert werden. Durch das Drücken der Taste 'f' soll der Nebel aktiviert bzw. deaktiviert werden. Mit der Taste 'i' soll bei aktiviertem Nebel zwischen den Modi GL_EXP, GL_EXP2 und GL_LINEAR umgeschaltet werden.

Aufgabe 20 (H) Texture Mapping

Sie können nun 3D Objekte zeichnen, Szenen modellieren und mit Licht versehen. Jedoch erscheinen die 3D Objekte in der Szene noch sehr unrealistisch, da selbst mit guter Lichtgebung sie doch lediglich Farben einer Farbpalette zugewiesen bekommen können. Um die Szene möglicherweise noch realistischer wirken zu lassen kann man Texturen verwenden.

Lesen Sie sich das Kapitel 9, 'Texture Mapping', im OpenGL Programming Guide (Red Book) durch. Texturen sind im Wesentlichen Bilder, die auf die Oberfläche von Strukturen (z.B. Polygone) abgebildet werden.

Um Bilder (Texturen) laden zu können, verwenden Sie die "imagedlib", indem Sie das VS.NET Project-File cvs/tools/bmpfiles/imagedlib.vcproj einbinden. Falls Sie unter Linux / UNIX programmieren wollen, können Sie das Makefile im Verzeichnis cvs/tools/bmpfiles/bmptools verwenden, um die Bibliothek zu erstellen. Wie schon aus den vorherigen Blättern bekannt können Sie mit der Klasse Bitmap eine .bmp-Datei laden und verwenden.

- Erweitern Sie die Bibliothek shape3D so, dass shapes auch Texturen auf ihre Oberflächen abbilden können. Versuchen Sie den Code möglichst überschaubar zu halten und durch Vererbung Kopieren von Code zu vermeiden. Um Konstruktoren von Superklassen aufzurufen verwenden Sie *Initializer Lists*.
- Bilden Sie zwei verschiedene Texturen auf Tetraeder und Quader aus dem vorherigen Aufgabenblatt ab.
- Da man diese 3D Objekte skalieren kann, versuchen Sie *mipmapping* zu verwenden, um bei bestimmten Skalierungsgrößen auf geringere oder höhere Auflösung der Texturen umzuschalten.

- Versuchen Sie mit geeigneten Texturen die Teapot-Szene realistischer zu machen. Manche Teekannen sind durchsichtig - versuchen Sie auch dies (mittels *blending*) zu realisieren.