

Computergrafik 1

Abgabetermin:

Die Lösung zu diesem Übungsblatt ist bis zum Freitag den **22. Mai 2009, 12:00 Uhr s.t.** über UniWorx abzugeben.

Inhalt:

Dieses Blatt dient zum Erlernen von grundlegenden Routinen zum Erstellen von Primitiven und komplexen Objekten in *OpenGL*. Hierzu werden Sie in Ihrem bereits erstellten Fenster nun verschiedene Arten der Objekterzeugung verwenden. In diesem Blatt erstellen Sie Objekte auf drei (aus gängigen Modellierungswerkzeugen bekannte) Weise: *Extrude* (verschieben der Grundform entlang einer Achse), *Lathe* (Rotation einer Form um eine Achse) und *Loft* (Bewegen einer Form entlang eines Pfades). Zusätzlich erzeugen Sie die Standard-Objekte *Quader*, *Pyramide*, *Kegel* und *Kugel*. Verwenden Sie in diesem Blatt **KEINE** glu-Hilfsfunktionen (z.B. `gluCylinder`, `gluSphere`, `gluNewQuadric`, etc.). Pro Aufgabe können maximal zehn Punkte erreicht werden. Zum Bestehen des Übungsblatts müssen in **JEDER** Aufgabe mindestens fünf Punkte erreicht werden.

Geben Sie insgesamt (d.h. ein Projekt für alle Aufgaben zusammen) alle benötigten Header-, Source-, und Projektdateien (von *Qt Creator* erzeugt) mit ab, d.h. *.h, *.cpp und *.pro Dateien. Abgaben die nicht kompilieren werden mit 0 Punkten bewertet. Fassen Sie alle Aufgaben zu einer zip-Datei zusammen und laden Sie sie bei UniWorx hoch.

Aufgabe 12 Einfache Primitive

(10 Punkte)

Diese Aufgabe dient zum Erlernen der Erzeugung einfacher Primitive. Um solche in *OpenGL* zu realisieren, können Sie die Standardfunktionen `glBegin`, `glEnd`, `glVertex` und `glColor` verwenden. Überlegen Sie sich genau, welchen Typ (z.B. `GL_LINES`, `GL_LINE_STRIP`, `GL_TRIANGLES`, `GL_QUADS`, etc.) Sie für eine performante Implementierung verwenden sollten.

- Erstellen Sie sich zunächst ein übergeordnetes Objekt `SceneObject`, das die für alle Objekte gültigen Parameter enthält. Dazu gehören die Position und Rotation eines Objektes, sowie dessen Farbe. Erstellen Sie Zugriffsfunktionen (`get` und `set`) für alle Parameter. Zusätzlich sollte das Objekt eine **virtuelle** Funktion `render` zur Verfügung stellen, die dann von allen Objekten überschrieben werden kann. Diese Funktion ist für das eigentliche Zeichnen (d.h. setzen der Vertices) verantwortlich. Erweitern Sie außerdem die Klasse `Scene` derart, dass diese eine Liste (**Hinweis:** `vector()` im Namensraum `std` ist hier hilfreich) von Objekten enthält, die nacheinander gezeichnet werden. Stellen Sie außerdem eine Funktion zur Verfügung, die es erlaubt Objekte der Szene hinzufügen zu können. Löschen Sie außerdem den Code zum Anzeigen des farbigen Würfels heraus (**Hinweis:** Beachten Sie, dass der Code für das Grid enthalten bleiben soll). (2 Punkte)
- Erstellen Sie die Grundprimitive *Quader*, *Pyramide*, *Kegel* und *Kugel* in jeweils einem Objekt, welches von `SceneObject` erbt. Jedes Objekt hat aber folgende individuelle Parameter, die zusätzlich zum `SceneObject` bereitgestellt werden müssen:

- *Quader*: Ein Quader hat drei Kantenlängen (Breite \times Höhe \times Tiefe). Daraus resultieren insgesamt sechs Seitenflächen (respektive acht Vertices).
- *Pyramide*: Eine Pyramide hat eine rechteckige Grundfläche und zusätzlich eine Höhe. Daraus ergeben sich insgesamt fünf Seitenflächen (Boden und vier Mantelflächen), sowie fünf Vertices.
- *Kegel*: Ein Kegel hat eine runde Fläche, sowie eine Höhe. Definieren Sie zusätzlich eine Anzahl N Seiten (d.h. die Qualität der Grundfläche). Übliche Werte sind hier 36 und mehr. Aus diesen Angaben resultieren dann $N + 1$ Seitenflächen (N Mantelflächen und eine Grundfläche), sowie $N + 1$ Vertices.
- *Kugel*: Eine Kugel hat kreisförmig angeordnete Vertices (und damit Flächen). Definieren Sie auch hier eine Anzahl von Segmenten (Anzahl = N). N beschreibt die Anzahl von Segmenten, die bei einer Umdrehung vorhanden sind. Dies gilt sowohl für die x - als auch für die y -Achse. Daher ergeben sich $N \cdot (N/2 - 1) + 2$ Vertices. Die Zahl berechnet sich folgendermaßen: Für eine Voldrehung (um die x -Achse) benötigen Sie N Vertices. Zusätzlich benötigen Sie bei $N/2$ Segmenten (nur 180° statt 360° auf der y -Achse gedreht) $N/2 - 1$ Vertices, da das letzte Segment jeweils auf den Pol der Kugel zeigt. Gerade wegen diesem Pol benötigen Sie zwei zusätzliche Punkte.

Erstellen Sie ein Objekt von jedem Typ und stellen Sie es (überschneidungsfrei) in Ihre Szene. Geben Sie allen Objekten zudem unterschiedliche Farben. (8 Punkte)

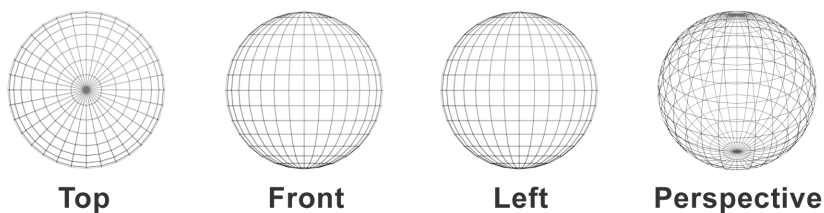


Abbildung 1: Eine Kugel mit 32 Segmenten

Aufgabe 13 *Extrude-Objekte*

(10 Punkte)

In dieser Aufgabe erstellen Sie nun komplexere Objekte. Zunächst soll hier das Extrudieren (d.h. das Verschieben der Grundform) erlernt werden. Die Teilaufgaben helfen Ihnen ein solches Objekt zu erzeugen.

- Erstellen Sie sich zunächst ein Objekt namens `ExtrudeObject`, das wiederum von `SceneObject` erbt. Dieses Objekt benötigt zusätzlich eine Achse (um die später verschoben werden soll), eine Grundform und eine Länge, um die verschoben werden soll. **Hinweis:** Als Grundfläche eignet sich eine einfache Punktliste, d.h. ein Array von drei-dimensionalen Punkten. (2 Punkte)
- Implementieren Sie nun die Funktion `render` wie folgt:
 - Berechnen Sie sich die ersten Punkte (d.h. alle Punkte der Form). Üblicherweise ist das die Position, an der die Form ist. Damit ist die Transformation die Einheits-Matrix.
 - Berechnen Sie sich zusätzlich den Mittelpunkt der Form. Summieren Sie dazu die Punkte, und teilen Sie dann durch die Anzahl der Punkte. Den Mittelpunkt benötigen Sie später, um die Deckflächen des Objekts zu zeichnen.

- Zeichnen Sie nun die erste Deckfläche (d.h. die Grundfläche) mit Hilfe des gerade berechneten Mittelpunktes.
- Berechnen Sie sich nun die neuen (d.h. verschobenen) Punkte der Form. Erstellen Sie die zugehörige Transformationsmatrix, die einen Punkt um die angegebene Achse mit der angegebenen Länge verschiebt. Wenden Sie dann diese Transformationsmatrix auf alle Punkte (d.h. Eckpunkte der Form sowie Mittelpunkt) an.
- Zeichnen Sie nun die Deckfläche mit Hilfe der gerade berechneten Punkte.
- Erstellen Sie nun noch die Seitenflächen, d.h. die Flächen, die sich zwischen neuen und alten Punkten befinden. (**Hinweis:** Dies können Sie sehr schnell mit `GL_QUADS` realisieren.) Achten Sie darauf, jeweils korrespondierende Punkte beider (also alter und neuer) Flächen zu verbinden.

Erstellen Sie ein *Extrude*-Objekt und stellen Sie es (überschneidungsfrei) in Ihre Szene. Geben Sie diesem Objekt wiederum eine unterschiedliche Farbe. **(8 Punkte)**

Aufgabe 14 *Lathe*-Objekte

(10 Punkte)

Eine weitere Form Objekte zu erstellen ist die sogenannte *Lathe*-Methode. Dabei wird eine zuvor definierte Form um eine gegebene Achse rotiert. Die Rotation muss jedoch nicht zwangsläufig volle 360° betragen. In dieser Aufgabe sollen Sie ein solches Objekt erzeugen. Die nachfolgenden Teilaufgaben sollen Ihnen dabei behilflich sein.

- a) Erstellen Sie sich zunächst ein Objekt namens `LatheObject`, das wiederum von `SceneObject` erbt. Dieses Objekt benötigt zusätzlich eine Achse (um die später gedreht werden soll), eine Form (die gedreht werden soll), die Rotationslänge (zwischen 0 und 360°) sowie eine Genauigkeit (d.h. wie oft soll die Rotationslänge unterteilt werden). Die Genauigkeit gibt hierbei wieder die Qualität an, d.h. je größer die Zahl, desto mehr Vertices, aber auch desto genauer (also „runder“) das Objekt. **Hinweis:** Als Form eignet sich eine einfache Punktliste, d.h. ein Array von drei-dimensionalen Punkten. Gleiches gilt für die Achse. **(2 Punkte)**
- b) Implementieren Sie nun die Funktion `render` wie folgt:
 - Berechnen Sie sich die ersten Punkte (d.h. alle Punkte der Form). Üblicherweise ist das die Position, an der die Form ist. Damit ist die Transformation die Einheits-Matrix.
 - Berechnen Sie sich nun die Punkte für eine Form, die um einen bestimmten Winkel um die gegebene Achse gedreht ist. Dazu berechnen Sie sich zunächst die Transformationsmatrix und wenden diese dann auf alle Punkte der Form an. Der Winkel ergibt sich aus der Division von gesamter Rotationslänge, sowie der Qualitätszahl.
 - Führen Sie den letzten Schritt für alle Segmente durch, d.h. mit der Qualitätszahl N genau N -mal.
 - Erstellen Sie nun mit den berechneten Punkten die Seitenflächen, d.h. die Flächen, die sich zwischen neuen und alten Punkten befinden. (**Hinweis:** Dies können Sie sehr schnell mit `GL_QUADS` realisieren.) Achten Sie darauf, jeweils korrespondierende Punkte beider (also alter und neuer) Flächen zu verbinden.
 - Erstellen Sie nun noch die Deckflächen. **Hinweis:** Verwenden Sie hierzu (anders als in einem Spezialfall möglich) auch `GL_QUADS`..

Erstellen Sie ein *Lathe*-Objekt und stellen Sie es (überschneidungsfrei) in Ihre Szene. Geben Sie diesem Objekt wiederum eine unterschiedliche Farbe. **(8 Punkte)**

Aufgabe 15 Loft-Objekte

(10 Punkte)

Eine dritte Form Objekte zu erstellen ist die sogenannte *Loft*-Methode. Dabei wird eine Grundform entlang eines zuvor definierten Pfades verschoben. Prinzipiell kann dies als eine Kette von *Extrude*-Objekten aufgefasst werden, wobei einige Dinge zu berücksichtigen sind. In dieser Aufgabe sollen Sie ein solches Objekt erzeugen. Die nachfolgenden Teilaufgaben sollen Ihnen dabei behilflich sein.

- a) Erstellen Sie sich zunächst ein Objekt namens `LoftObject`, das wiederum von `SceneObject` erbt. Dieses Objekt benötigt zusätzlich einen Pfad (um die später verschoben werden soll) und eine Form (die verschoben werden soll). **Hinweis:** Als Form eignet sich eine einfache Punktliste, d.h. ein Array von drei-dimensionalen Punkten. Gleiches gilt für den Pfad. (2 Punkte)
- b) Implementieren Sie nun die Funktion `render` wie folgt:

- Berechnen Sie sich die ersten Punkte (d.h. alle Punkte der Form). Üblicherweise ist das die Position, an der die Form ist. Damit ist die Transformation die Einheits-Matrix.
- Berechnen Sie sich nun die Punkte für jede Form an einem neuen Vertex des Pfades, d.h. bei einer Pfadlänge von N Punkten müssen Sie nun noch $N - 1$ Formen (bzw. deren Punkte) berechnen. Dies funktioniert analog zum *Extrude*-Objekt, muss aber mehrmals ausgeführt werden.
- Üblicherweise sind die Pfade nicht linear, d.h. zwischen einzelnen Pfadsegmenten existiert ein Winkel $\neq 0$. Daher muss die Grundform auch entsprechend gedreht werden. Die Drehung ist allerdings für alle Mittelelemente (also von Punkt 2 bis Punkt $N - 1$ bei N Punkten nur die Hälfte des tatsächlichen Winkels zwischen den Segmenten. Den Winkel zwischen zwei Vektoren bestimmen Sie mit Hilfe folgender Formel:

$$\cos(\alpha) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|};$$

Die Drehachse (die auch direkt bei `glRotate` angegeben werden muss) kann über die Normale auf die beiden Segmentvektoren ermittelt werden. Die Normale ermitteln Sie wie folgt:

$$\vec{n} = \vec{a} \times \vec{b} \text{ (wobei } \times \text{ das Vektorprodukt ist!)}$$

- Erstellen Sie nun die Deckflächen, d.h. die Fläche an bei dem Pfadpunkten 1 und N (**Achtung:** Die Zählung beginnt hier bei 1!). Dazu benötigen Sie wiederum den Mittelpunkt, der wie in Aufgabe 13 beschrieben berechnet werden kann.
- Zeichnen Sie nun noch die Seitenflächen des *Loft*-Objekts. Verwenden Sie dazu die in Aufgabe 13 beschriebene Methode - auch hier wieder mehrmals, d.h. für jeden Abschnitt.

Erstellen Sie ein *Loft*-Objekt und stellen Sie es (überschneidungsfrei) in Ihre Szene. Geben Sie diesem Objekt wiederum eine unterschiedliche Farbe. Der Pfad darf hierbei nicht linear sein, d.h. zwischen zwei aufeinander folgenden Segmenten muss ein Winkel $\neq 0$ existieren. Der Einfachheit halber dürfen Sie den Pfad jedoch in einer Ebene lassen (d.h. eine der drei Koordinatenachsen ist 0 für jeden Punkt). (8 Punkte)