

3D Programmierpraktikum

Abgabetermin:

Die Lösung zu diesem Übungsblatt ist bis zum Montag den **14. Mai 2007, 12:00 Uhr s.t.** abzugeben.

Inhalt:

Dieses Übungsblatt dient zur Einführung in die Grafikprogrammierung mit *GLUT* und *OpenGL*. Es werden Grundlagen zum Erzeugen von Fenstern mit GLUT sowie das Rendern von 3D Inhalten mittels OpenGL vermittelt. Außerdem wird darauf eingegangen, wie Benutzereingaben mit GLUT verarbeitet werden.

Aufgabe 6 (P) Ein Fenster mit GLUT

In dieser Aufgabe erstellen Sie ein Fenster mit dem *OpenGL Utility Toolkit (GLUT)*. Der hier erstellte Code sollte möglichst für alle weiteren Aufgaben wiederverwendbar sein.

- a) Laden Sie sich die Binaries von *GLUT* für Windows von der folgenden Seite herunter und kopieren Sie die in dem Archiv enthaltenen Dateien wie in der Datei *README-win32.txt* beschrieben: <http://www.xmission.com/~nate/glut.html>
- b) Erstellen Sie ein eigenes C++-File für die main-Methode. Stellen Sie eine Möglichkeit zur Verfügung, die es erlaubt, die Breite und Höhe des Fensters *optional* als Parameter zu übergeben, also z.B. `aufgabe_6 1024 768`. Initialisieren Sie *GLUT* bereits hier mit den Funktionen `glutInit(int*, char**)` und `glutInitDisplayMode(unsigned int)`.
- c) Erstellen Sie je ein Header-File sowie ein C++-File namens `Controller.*` für die Hauptfunktionen von GLUT (siehe Folien).
- d) Erstellen Sie nun eine Klasse namens *View* für das eigentliche Fenster. Diese sollte folgende Eigenschaften besitzen:
 - **Attribute:** Titel, Fenster-ID, Breite, Höhe und ein `bool`, der angibt, ob sich das Fenster gerade im Vollbildmodus befindet.
 - **statische Attribute** `viewCounter`, ein Zähler der angibt um das wie viele Fenster es sich handelt.
 - **Funktionen:**
 - Geeigneter Konstruktor und Destruktor
 - Initialisierung des Fensters (GLUT-Funktionen anbinden)
 - Änderung der Breite und Höhe
 - Ändern des Display-Modus (Fullscreen oder Window)

- e) Fügen Sie dem Controller eine Funktion namens `createWindow(int, int)` hinzu, die die Breite und die Höhe des zu erzeugenden Fensters als Parameter erwartet. In dieser Methode sollen nun folgende Schritte durchgeführt werden:
- Erstellen einer Instanz der Klasse *Window*
 - Initialisierung von *OpenGL* (siehe Folien)
 - Starten der Funktion `glutMainLoop()`
- f) Implementieren Sie die Callback-Funktionen für `glutDisplayFunc`, `glutIdleFunc` und `glutReshapeFunc`, so dass ein Dreieck auf schwarzem Hintergrund angezeigt wird.
- g) Implementieren Sie eine Callback-Funktion für `glutKeyboardFunc`, so dass das Programm beim Drücken der ESC-Taste beendet wird.
- h) Implementieren Sie zudem eine Callback-Funktion für `glutSpecialFunc` damit das Fenster beim Drücken der F1-Taste zwischen Vollbildmodus und Fenster-Modus wechselt.

Aufgabe 7 (P) Einfache 2D-Objekte

In dieser Aufgabe erstellen Sie einfache zweidimensionale Objekte in *OpenGL*. Verwenden Sie dazu den gesamten Code aus Aufgabe 7 für die grafische Ausgabe.

- a) Erstellen Sie eine Klasse *Form* mit folgenden Eigenschaften:
- Attribute für den Zustand (*aktiv* oder nicht), und die aktuelle Rotation um die *x*-, *y*- und *z*-Achse (drei Werte)
 - Getter- und Setter-Funktionen für *alle* Attribute sowie eine abstrakte Funktion namens *renderForm()*
- b) Erstellen Sie die zwei verschiedenen Formen *SquareShape* und *OctagonShape*, die als Parameter die Länge einer Kante (als `float`) erwarten. Beide Klassen sollen von *Form* erben und die abstrakte Funktion *renderForm()* implementieren.
- c) Die Formen sollen im *inaktiven* Zustand einen hellgrauen Farbton besitzen. Sobald eine Form aktiv ist, soll jeder Vertex (beim Achteck jedes Vertex-Paar) eine unterschiedliche Farbe erhalten.
- d) Erweitern Sie Ihr Programm um die nachfolgend genannten Tastenfunktionen:
- **W**: Wechselt das aktive Objekt (d.h. zu Beginn muss eins der beiden Objekte *aktiv* sein). Beim Drücken der Taste wird das aktive Objekt *inaktiv* und das inaktive Objekt *aktiv*
 - **Pfeiltasten (hoch bzw. runter)**: Rotation um die *x*-Achse des Objekts
 - **Pfeiltasten (links bzw. rechts)**: Rotation um die *y*-Achse des Objekts
 - **Bildlauf (auf bzw. ab)**: Rotation um die *z*-Achse des Objekts
 - **Leertaste**: Rücksetzen der Szene in den Anfangszustand (d.h. alle Rotationen auf null)

Hinweis: Beachten Sie, dass sich Rotationen nur auf das jeweils *aktive* Objekt bezieht. Das *inaktive* Objekt behält seine gegenwärtige Transformation bei, d.h. für den Fall, dass es bereits rotiert wurde, soll diese Transformation beim Rendern beibehalten werden. Die Leertaste hingegen wirkt sich auf *beide* Objekte aus.

Aufgabe 8 (P) Einfache 3D-Objekte

In dieser Aufgabe erstellen Sie einfache dreidimensionale Objekte in *OpenGL*. Verwenden Sie dazu den gesamten Code aus Aufgabe 7 für die grafische Ausgabe.

- a) Erstellen Sie eine Klasse *CubeShape* mit folgenden Eigenschaften:
- Ein Attribut für die Länge einer Kante des Würfels
 - Drei Attribute für die Rotation um eine der drei Achsen
 - Drei Attribute für die Rotationsgeschwindigkeit um eine der drei Achsen
 - Ein Attribut für die Position des Würfels
 - Funktion für das Rendern des Würfels
 - Funktionen für das Ändern der Rotationen des Würfels
 - Funktionen für das Ändern der Rotationsgeschwindigkeiten des Würfels
 - Funktion für das Bewegen des Würfels
- b) Geben Sie jeder Fläche des Würfels beim Rendern eine unterschiedliche Farbe.
- c) Erweitern Sie Ihr Programm um folgende Tastatur-Funktionen:
- **Pfeiltasten (hoch bzw. runter)**: Erhöhung bzw. Verringerung der Rotationsgeschwindigkeit um die x -Achse des Objekts
 - **Pfeiltasten (links bzw. rechts)**: Erhöhung bzw. Verringerung der Rotationsgeschwindigkeit um die y -Achse des Objekts
 - **Bildlauf (auf bzw. ab)**: Erhöhung bzw. Verringerung der Rotationsgeschwindigkeit um die z -Achse des Objekts
 - **+** : Bewegen des Würfels auf den Betrachter zu
 - **-** : Bewegen des Würfels auf den Betrachter zu
 - **Leertaste**: Rücksetzen der Szene in den Anfangszustand (d.h. alle Rotationen auf null)
 - **Returntaste**: Stoppen aller Rotationen, d.h. die Rotationsgeschwindigkeiten auf null setzen