Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Andreas Butz
Hendrik Richter, Raphael Wimmer

Computergrafik 1
Sommersemester 2010
Übungsblatt 4

Übungsblatt 4: OpenGL-Basics

Abgabe:
Dieses Übungsblatt ist als Gruppe zu lösen. Die Lösung ist bis **Dienstag,** den **1. Juni 2010, 12:00 Uhr s.t.** über UniWorx (http://www.pst.ifi.lmu.de/uniworx) abzugeben.
Es werden nur die Formate PDF und Plain-Text (UTF-8) akzeptiert. Erstellen Sie für jede Aufgabe ein Unterverzeichnis nach dem Schema <Übungsblatt>-<Aufgabe>, d.h. die Lösung der ersten Aufgabe kommt in ein Verzeichnis 4-*1/*. Packen Sie alle Dateien in eine ZIP-Datei und laden Sie diese bei UniWorx hoch. Wenn Sie Formatierungsvorgaben nicht einhalten, werden bis zu zwei Punkte abgezogen. Lösungen müssen zumindest im CIP-Pool fehlerfrei kompilieren und laufen. Bitte geben Sie nur Quellcode ab, keine kompilierten Dateien.

Inhalt:
Ziel dieses Übungsblattes ist, den grundlegenden Aufbau von OpenGL zu verstehen, sowie die wichtigsten Elemente der API kennenzulernen.
Wir empfehlen, die OpenGL-Beispiele von QT anzuschauen, und sich von dort Anregungen zu holen. Desweiteren gibt es im WWW eine Vielzahl an OpenGL-Tutorials.
Verwenden Sie zum Zeichnen den “immediate mode”, wenn nichts anderes gefordert wird.
Verwenden Sie nur Funktionen aus OpenGL, QT und der STL. Verwenden Sie z.B. nicht GLUT.

Es können maximal 20 Punkte erreicht werden.

Aufgabe 1: OpenGL-Pipeline (5 Punkte)

Lesen – und verstehen – Sie eine Beschreibung der OpenGL-Pipeline, z.B. [1]. Dieser Überblick wird in allen folgenden Übungsblättern vorausgesetzt. Verwenden Sie die auf opengl.org verfügbaren Dokumente als Referenz für die Bearbeitung der Übungsaufgaben.

a) Beschreiben Sie in eigenen Worten, was Backface Culling (aka Face Culling) ist, und weshalb dessen Verwendung im Allgemeinen Performancevorteile bringt. Wie ist Backface Culling in OpenGL implementiert?
b) Sie sollen ein Computerspiel implementieren, in dem hunderte von ähnlich, aber nicht gleich, aussehenden Häusern gleichzeitig zu sehen sind. Welche Methode, Primitive zu zeichnen, wäre hier besonders ineffizient, welche besonders effizient? Begründen Sie Ihre Entscheidung kurz.

[1] http://www.opengl.org/wiki/Rendering_Pipeline_Overview

Geben Sie eine Datei opengl.txt oder opengl.pdf im Unerverzeichnis 4-1/ ab.

Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Andreas Butz
Hendrik Richter, Raphael Wimmer

Aufgabe 2: Primitive (5 Punkte)

Erzeugen Sie eine QT-Anwendung, die ein OpenGL-Widget enthält.
In diesem sollen 8 Primitive in einer 2x4-Anordnung gezeichnet werden. Die Objekte sollen komplett und möglichst groß zu sehen sein . In jeder Zeile sollen ein Würfel und eine Pyramide mit gleicher Grundfläche und Höhe gezeichnet werden. Verwenden Sie für das Zeichnen der ersten Zeile den *immediate mode*, für die zweite Zeile Display Lists, für die dritte Zeile ein Vertex Array und für die vierte Zeile ein Vertex Buffer Object (VBO). Versuchen Sie, die Zeichenmethoden möglichst effizient einzusetzen.
Geben Sie ein Projekt *primitive.pro* inkl. Quellcode und Headern in einem Unterverzeichnis *4-2/* ab.

Aufgabe 3: 3D-Transformationen (5 Punkte)

Erzeugen Sie eine QT-Anwendung, die ein OpenGL-Widget enthält. Im OpenGL-Widget wird eine Uhr angezeigt. Diese besteht aus einem Stundenzeiger, einem Minutenzeiger, einem Sekundenzeiger und einem Ziffernblatt. Das Ziffernblatt enthält mindestens 12 Markierungen für die Stunden. Die Zeiger und Markierungen können z.B. als einfache Boxes oder auch als Flächen gezeichnet werden. Zeichnen Sie alle Objekte im Ursprung und transformieren Sie sie an die passenden Stellen. Die Uhr soll immer die aktuelle Zeit anzeigen. Dazu können Sie `QTime::currentTime()` verwenden. Geben Sie den Zeigern und dem Ziffernblatt unterschiedliche, übliche Farben.
Geben Sie ein Projekt *transform.pro* inkl. Quellcode und Headern in einem Unterverzeichnis *4-3/* ab.

Aufgabe 4: Kamera (5 Punkte)

Erzeugen Sie eine QT-Anwendung, die ein OpenGL-Widget enthält.
Im Mittelpunkt sollen zwei Würfel nebeneinander angezeigt werden. Bei gedrückter linker Maustaste soll sich die Kamera – bei konstantem Abstand - um die Objekte herum rotieren lassen. Siehe dazu auch das HelloGL-Beispiel von QT, dessen Quellcode Sie für diese Aufgabe verwenden dürfen. Bei gedrückter rechter Maustaste soll die Kamera bei einer vertikalen Mausbewegung eine Fahrt auf die Objekte zu, bzw. von ihnen weg, machen.
Mit dem Scrollrad soll sich der Öffnungswinkel der Kamera in sinnvollen Schritten ändern lassen. Durch Drücken der mittleren Maustaste wird zwischen perspektivischer und orthogonaler Projektion gewechselt.
Geben Sie ein Projekt *camera.pro* inkl. Quellcode und Headern in einem Unterverzeichnis *4-4/* ab.

Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Andreas Butz
Hendrik Richter, Raphael Wimmer

Tipps zum Arbeiten in der Gruppe:

- Richten Sie ein gemeinsames Repository (SVN, Git) ein, in dem Sie Ihre Lösungen verwalten. Achten Sie darauf, dass es nur für die Gruppe les- und schreibbar ist. Infos dazu auf der Seite der Rechnerbetriebsgruppe.
- Verwenden Sie eine Mailingliste (kann man im CIPconf einrichten) oder andere Kommunikationswege (z.B. Google Wave) um mit Gruppenmitgliedern zu kommunizieren.
- Wenn möglich, bearbeiten Sie Teilaufgaben zu zweit in wechselnder Zusammenstellung. Dadurch lernen Sie am meisten und halten alle Gruppenmitglieder auf dem gleichen Stand. Siehe dazu auch Tipps zum Pair Programming im WWW.
- Die einzelnen Aufgaben eines Übungsblattes überschneiden sich oft. Programmieren Sie gemeinsame Funktionalität zusammen und informieren Sie Teammitglieder, wenn Sie ein Problem gelöst haben, das für diese relevant sein könnte. Innerhalb der Gruppe ist das wiederverwenden von Code erlaubt und gewünscht.
- Beachten Sie, dass neue Uploads bei UniWorx die alten komplett überschreiben. Geben Sie daher immer nur vollständige Lösungen ab.
- Geben Sie bei jeder Abgabe bitte die CIP-Kennungen aller Gruppenmitglieder in UniWorx an.
- Wenn ein Gruppenmitglied aussteigen will, bitte eine E-Mail an die Übungsleiter schicken.
- Lesen und beherzigen Sie Randy Pauschs Tipps zur erfolgreichen Gruppenarbeit: http://www.cl.cam.ac.uk/teaching/GroupProjects/Pausch-tips.pdf

Viel Erfolg.