

Systemdokumentation - Computergrafik 1

Clemens Dehnert

Januar, 2024

Inhaltsverzeichnis

1	Aufgabenstellung	3
2	Lösungsansatz	3
3	Lösungsumsetzung	3
4	Installations- und Bedienungsanleitung	4
4.1	Installation	4
4.2	Bedienung	4
5	Screenshots	5
6	Probleme	8
7	Ergebnisse	8
8	Literatur- und Quellverzeichnis	9

1 Aufgabenstellung

Im Rahmen des Moduls Computergrafik 1 im Studiengang Allgemeine Informatik, war ein Programm mit OpenGL und eine dazugehörige Systemdokumentation zu entwerfen. Die Abnahme der Komplexaufgabe dient als Zulassung zur schriftlichen Prüfung. Dabei sollten gewonnene Kenntnisse aus den vorangegangenen Praktika und der Vorlesung angewandt werden. Ziel war es eine Szene mit verschiedenen dreidimensionalen Objekten samt Animation und unterschiedlichen Lichtquellen zu erstellen. Dabei sollte die Betrachtung aus mehreren Perspektiven und mit verschiedenen Projektionen möglich sein. Anbei sollte die hier vorliegende Dokumentation abgegeben werden, welche den Lösungsweg zur Programmieraufgabe darlegt.

2 Lösungsansatz

Den Lösungsansatz stellten die vorangegangenen Praktika dar, aus welchen der Quellcode entsprechend adaptiert werden sollte. Des Weiteren sollte die Abstraktion der Objekte als Klassen die Wartbarkeit des Codes und Skalierbarkeit des Projektes wesentlich verbessern. Dies gelang zwar auch, allerdings führte diese Herangehensweise zu einem erheblichen Verlust an Flexibilität. Genaues wird dazu in der Sektion Probleme beschrieben. Bei der Bearbeitung der Aufgabe wurde inkrementell vorgegangen.

3 Lösungsumsetzung

Die Erarbeitung des Endresultates erfolgte über einen langen Zeitraum. Zu Beginn konnten viele Techniken aus den vorangegangenen Praktika angewandt werden. Die Zusammensetzung erwies sich größtenteils als einfach. Die Parametrisierung der in Klassen abstrahierten Objekte sorgen für unnötigen Aufwand. Zunächst wurden die Shader nicht weiter als nötig ausgearbeitet. Stattdessen wurde der GLManager erstellt, welcher als Singleton allen Objekten, die Memberdaten von ihm benötigten, diese bereitstellte. Die Klasse Camera sollte die Steuerung wesentlich vereinfachen. Danach wurde der Fragmentshader um die Lichtpunkte und das Richtungslicht erweitert. Zum Ende des Projektes wurden die Viewports eingerichtet und die Darstellungen entsprechend modifiziert. Final musste das Projekt auf Windows portiert und die Systemdokumentation erstellt werden. Die Portierung erfolgt an den Windowsrechnern in der HTW Dresden.

4 Installations- und Bedienungsanleitung

4.1 Installation

Es kann entweder einfach die .exe-Datei ausgeführt werden oder unter Verwendung von Visual Studio die .exe generiert werden, welche dann standardmäßig unter dem Ordner Debug zu finden ist. Dazu müssen jedoch die notwendigen Bibliotheken installiert sein. Dabei handelt es sich um glm, glew und freeglut.

4.2 Bedienung

Die Kameraposition kann unter der Verwendung der Tasten W vorne, S für hinten, A für links und D für rechts angepasst werden. Zudem verändern die Tasten Q und E die Höhe, auf welcher sich die Kamera befindet. Es ist auch die Funktionalität implementiert, welche er dem Nutzer erlaubt sich mit der Maus umzusehen. Zudem kann die Geschwindigkeit des Riesenrades im Hintergrund mittels der Tasten R beziehungsweise G verändert werden. Die Größe des Fensters kann beliebig verändert werden.

5 Screenshots

In diesem Abschnitt wird das finale Programm dargestellt und Fehlversuche, die bei der Erstellung aufgetreten sind.

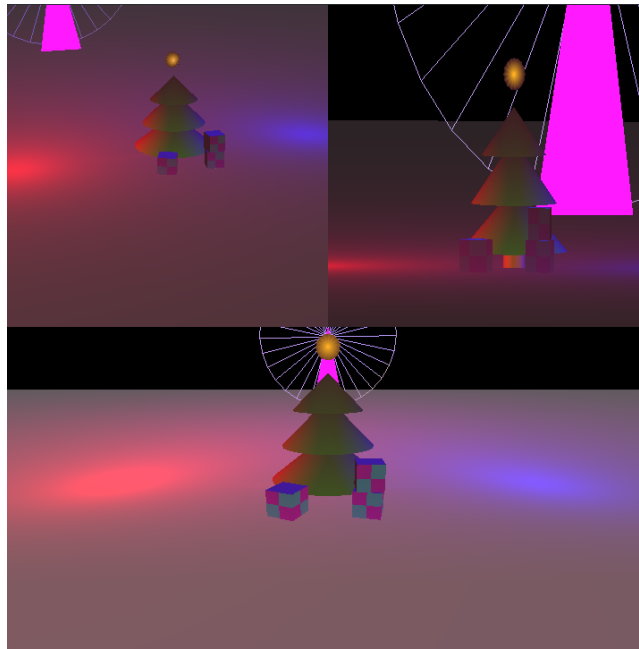


Abbildung 1: Ansicht nach Programmstart

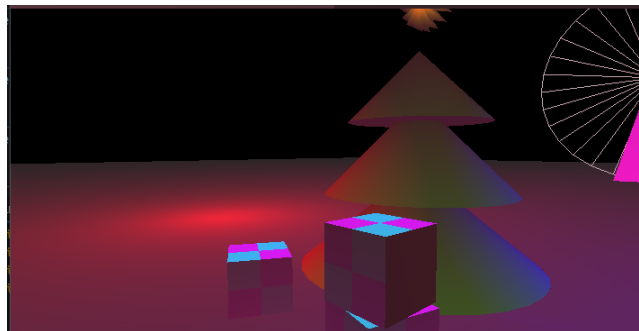


Abbildung 2: Auswirkung des Richtungslichts



Abbildung 3: Auswirkung des Richtungslichts

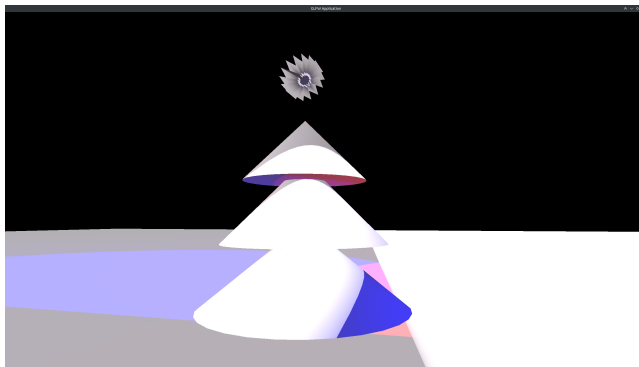


Abbildung 4: Experimentieren mit dem Shader

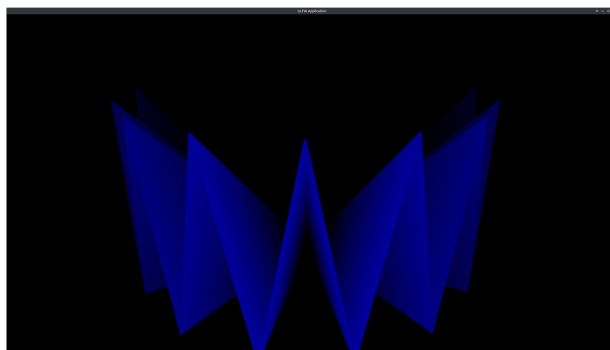


Abbildung 5: Fehlversuch bei der Erstellung eines Zylinders

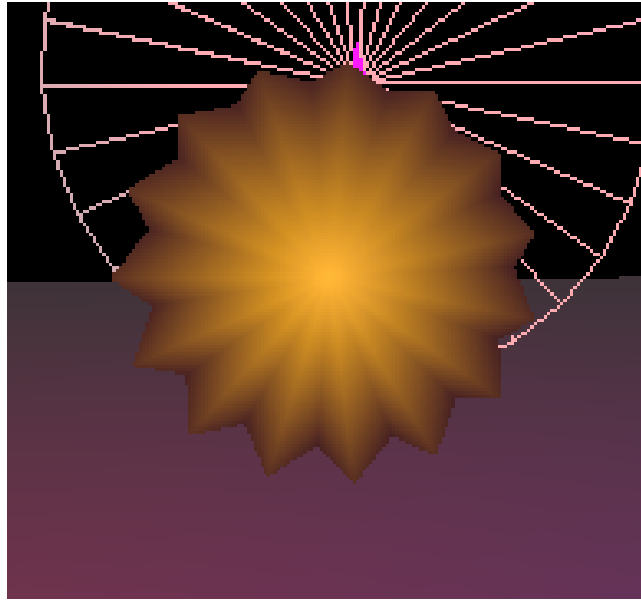


Abbildung 6: Nutzung des Fehlversuches als Stern

6 Probleme

Im Allgemeinen taten sich zwei größere Probleme auf. Zum einen waren manche, der auf den Windowsrechnern angewandten Tools nicht unter Linux verfügbar - beziehungsweise zu stark veraltet. Allerdings konnte hier die GLFW Bibliothek hier Abhilfe schaffen. Bis auf einige andere Namen der Callback-Funktionen ließ sich das Projekt gut umsetzen. Abschließend musste der Code nur noch auf einem Windowsrechner übertragen und dort entsprechen abgeändert werden. Das zweite größere Problem, war die Entscheidung mit Klassen zu arbeiten. Die Arbeit in Klassen erbrachte nicht die erhofften Vorteile bei der Bearbeitung der Komplexaufgabe. Die Organisation der Objekte war recht leicht, aber Modifikationen für spezielle Objekte einer Klasse nicht möglich, wenn die Parameterliste Konstruktoren nicht ausarten sollten. Das führte dazu, dass die Zahl an Quelldateien und Headerdateien wuchs, derselbe Code häufig an verschiedenen Stellen auftauchte und das Projekt an Bearbeitungsaufwand zunahm. Mittels einiger Features von C++, wie beispielsweise der Vererbung hätten einige der Codedopplungen möglicherweise vermieden werden können, allerdings hätte das den Fokus vollends von OpenGL auf die Programmieretechniken mit C++ verschoben. Ursprünglich sollten sich die Farben des Riesenrades ändern. Allerdings lässt OpenGL es nicht zu, dass das Array, welches die vertices enthält, nachträglich verändert wird. Mit dem Fragmentshader und einem Boolean als Uniformvariablen und einer Uniformvariable, welche zeitabhängig ist, hätte sich das Problem im Fragmentshader lösen lassen, hätte jedoch den Code immens verkompliziert.

7 Ergebnisse

Es liegt ein Programm vor, welches dieselbe Szene unterschiedlich zeigt. In der linken oberen Ecke ist die Szene mittels perspektivischer Projektion dargestellt. Die Position der Kamera verändert sich nicht. Im Gegensatz dazu steht das obere rechte Viertel, wo die orthogonale Projektion genutzt wird. Auch hier ist die Kamera unveränderlich. In der unteren Bildschirmhälfte findet eine perspektivische Projektion statt. Zudem kann die Kamera mittels der in der Anleitung beschriebenen Möglichkeiten verändert werden. Die Bildschirmmausschnitte skalieren auch bei einer Veränderung der Fenstergröße korrekt. Es wird ein Weihnachtsbaum mit drei Geschenken davor dargestellt. Im Hintergrund ist ein sich drehendes Riesenrad zu sehen. Zwei Punktlichter links und rechts vom Weihnachtsbaum beleuchten die Szene rot bzw. blau. Hinzu kommt ein Richtungslicht, dass von der Kameraausrichtung abhängt und ein dezentes Ambient-Licht. Wesentliche Lernergebnisse waren das Festigen des Umgangs mit OpenGL, die Verwendung von Latex für die Dokumentation und das sich selbstständige Aufbauen von Verständnis für ein komplexeres OpenGL-Projekt.

8 Literatur- und Quellverzeichnis

Etay, Meiri: ogldev, in: GitHub, 2021, https://github.com/emeiri/ogldev/tree/master/tutorial22_youtube (zuletzt abgerufen am 15.01.2024).

Vorlesungs- & Praktikumsunterlagen