

Computergrafik 1 (BIN)

Projektarbeit

Prof. Dr. Dennis Allerkamp – Sommersemester 2025

Stand: 19. Mai 2025

1 Ablauf

- Sowohl die Projektarbeit als auch die Klausur müssen für den erfolgreichen Abschluss der Lehrveranstaltung bestanden werden.
- Die Klausur findet am 23. Juni 2025 statt. Im September 2025 wird es eine Wiederholungsklausur geben.
- Die Abgabe der Projektarbeit muss spätestens bis zum 20. Juni 2025 über Moodle erfolgen. Wenn das gesamte Team die Klausur erst im September schreiben möchte, ist eine Abgabe bis dahin auch möglich.
- Nach der Klausur müsst ihr mir eure Programme präsentieren. Die Terminvergabe wird über Moodle organisiert. Die Präsentationen werden voraussichtlich online stattfinden. Bitte klärt vorab, wer von euch den Bildschirm teilen wird, um mir den Quellcode und das lauffähige Programm zu zeigen. Mehr müsst ihr dafür nicht vorbereiten.

2 Aufgaben

2.1 Organisation

- Klärt in eurem Team grundsätzliche Fragen der Zusammenarbeit:
 - Wann und wie könnt ihr euch gegenseitig am besten erreichen?
 - Wann trifft ihr euch für gemeinsame Absprachen?
 - Wer kümmert sich um welche Aufgabe?
- Richtet ein gemeinsames Git-Repository ein (z. B. bei <https://gitlab.gwdg.de>).
- Fügt dem Repository eine README-Datei mit den Namen der beteiligten Gruppenmitglieder hinzu.

2.2 Basisfunktionen

- Entwickelt Datentypen und Funktionen für alle benötigten Matrixfunktionen.
- Schreibt ein Testprogramm, das alle Matrixfunktionen gründlich testet. Beachtet dabei auch mögliche Sonderfälle (z. B. dass zwei Parameter auf dieselben Arrays verweisen). Das Testprogramm soll keine Werte ausgeben, sondern nur das Ergebnis von jedem Test (Testname, „OK“/„Nicht OK“).
- Entwickelt weitere nützliche Datentypen und Hilfsfunktionen für euer Demoprogramm (z. B. für Shaderprogramme, Texturen, Geometriedaten, ...).

2.3 Demoprogramm

- Einigt euch auf eine oder mehrere ansprechende 3D-Szenen für euer Programm, mit der ihr die geforderten Techniken (siehe Anforderungen) demonstrieren möchtet. Beschreibt die Szene bzw. die Szenen und die dort verwendeten Techniken kurz in der README-Datei.
- Erstellt für eure Szenen eigene Geometriemodelle und Texturen oder sucht welche aus dem Internet zusammen und gebt die Quellen in eurer README-Datei an.
- Implementiert die Szene bzw. die Szenen in einem Demoprogramm unter Verwendung der von euch entwickelten Basisfunktionen.

3 Anforderungen

- Als Projektarbeit soll mit OpenGL ein Demoprogramm für die in der Lehrveranstaltung behandelten Themen erstellt werden. Dafür sollt ihr eine oder mehrere ansprechende 3D-Szenen gestalten.
- Es wird kein produktreifes Ergebnis erwartet. Die Anwendung sollte jedoch in sich rund und intuitiv bedienbar sein.
- Die Anwendung muss auf den Pool-PCs funktionieren. (Vor der Abgabe solltet ihr das noch einmal testen.)
- Der Code soll umfassend dokumentiert werden, d. h. für jede Funktion soll es eine kurze Beschreibung geben.
- Alle Team-Mitglieder müssen in der Lage sein, den Code und die Funktionsweise der eingesetzten Techniken zu erklären.
- Der gesamte Code muss selbst entwickelt werden. Das Kopieren fremden Codes wird als Täuschungsversuch gewertet und führt automatisch zum Nichtbestehen. Code, der von euch selbst im Rahmen der Übungen entwickelt wurde, darf natürlich verwendet werden.
- Es dürfen nur die Standardbibliothek, GLEW, GLFW sowie eine Bibliothek zum Laden von Bilddaten eingebunden werden. Die Verwendung weiterer Bibliotheken ist nur nach Rücksprache möglich.
- Die Verwendung von fremden Texturen und Geometriedaten ist gestattet. Die Quellen sollen allerdings in der README-Datei angegeben werden.
- Achtet bei der Abgabe darauf, dass diese das Größenlimit von 100 MB einhält und keine unnötigen Dateien enthält (z. B. erzeugte Binärdateien oder das Git-Verzeichnis).

Folgende Mindestanforderungen sind zu erfüllen:

- README-Datei mit Informationen zum Bauen, zum Starten und zur Bedienung des Programms
- Matrixfunktionen zur Umsetzung der benötigten Transformationen (Verschiebung, Drehung, Skalierung, Position und Orientierung der Kamera, Perspektivprojektion, Normalenmatrix)
- Import von Geometriedaten aus separaten Dateien
- mehrere Objekte in einer 3D-Szene
- Animation der Kamera und/oder eines Objekts
- Beleuchtung der Szene, wie in der Vorlesung beschrieben
- Texturierung von mindestens einem Objekt
- einfache Benutzerinteraktion mit Kamera und/oder Objekten (z. B. über Tastatur)

Wenn ihr nicht nur bestehen sondern auch eine gute Note bekommen möchtet, müsst ihr auch möglichst viele der folgenden Anforderungen erfüllen:

- sinnvolle Aufteilung des Codes auf mehrere Dateien (insb. Shader-Code in separaten Dateien)
- Datentypen und Hilfsfunktionen für Shader, Geometriedaten, Texturen
- Datentypen zur Strukturierung der Szene
- Überprüfung übergebener Funktionsparameter durch Assertions (soweit möglich)
- Tests für alle Matrixfunktionen
- Beleuchtung der Szene mit mehr als einer Lichtquelle
- gleichzeitige Verwendung von mindestens zwei Texturen in einem Objekt
- gleichzeitige Verwendung von mindestens zwei verschiedenen Shader-Programmen in einer Szene
- transparente Objekte
- Verwendung einer Umgebungstextur
- Spiegelung der Umgebungstextur auf einigen Objekten
- Nebel in einer komplexen 3D-Szene

Für eine besonders gute Note ist die Umsetzung mindestens einer fortgeschrittenen Technik erforderlich, die nicht in der obigen Liste genannt wird und für die ihr die notwendigen Details selbst recherchieren müsst.

Weitere Bewertungskriterien sind Code-Qualität, Projektidee und Ästhetik.