

# Performante Bildsynthese

## Projektarbeit

Prof. Dr. Dennis Allerkamp – Wintersemester 2024/2025

Stand: 26. November 2024

### 1 Ablauf

- Die Prüfung besteht aus einer Projektarbeit und einer mündlichen Prüfung.
- Die Abgabe der Projektarbeit muss spätestens bis zum 20. Januar 2025 über Moodle erfolgen.
- Am 21./22. Januar 2025 finden die Präsentationen eurer Programme und die mündlichen Prüfungen statt. Die Terminvergabe wird über Moodle organisiert.
- Die Präsentation muss nicht weiter vorbereitet werden. Es reicht, wenn ihr mir kurz euer Programm und den dazugehörigen Quellcode zeigen könnt.
- Die mündlichen Prüfungen finden jeweils direkt im Anschluss an die Präsentation statt. Dabei werde ich einige Fragen zu eurem Programm und zum Thema der Vorlesung stellen.

### 2 Aufgabe

Im Rahmen der Projektarbeit sollt ihr ein Demoprogramm erstellen, das die in der Vorlesung behandelten Techniken zur Bildsynthese mit Vulkan demonstriert. Dafür sollt ihr eine ansprechende 3D-Szene gestalten, die die im Folgenden aufgeführten Anforderungen erfüllt.

Bewertungskriterien sind die Umsetzung der Anforderungen, die Code-Qualität, die Projektidee und die Ästhetik der Umsetzung.

#### 2.1 Allgemeine Anforderungen

- 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 nichttriviale Funktion und für alle Klassen 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. Die Verwendung von fremden Texturen und Geometriedaten ist gestattet.
- Es dürfen nur die Standardbibliothek, GLFW, GLM sowie Bibliotheken zum Laden von Geometrie- und Bilddaten eingebunden werden. Die Verwendung weiterer Bibliotheken ist nur nach Rücksprache möglich.
- 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).

#### 2.2 Mindestanforderungen

Euer Programm soll mindestens die folgenden Anforderungen erfüllen:

- README-Datei mit einer Beschreibung der Szene, mit Informationen zum Bauen, zum Starten und zur Bedienung des Programms sowie Quellenangaben der verwendeten Texturen und Geometriedaten
- Erzeugung einer Vulkan-Instanz unter Angabe der gewünschten Erweiterungen
- Auswahl eines geeigneten Geräts und passender Queue-Familien

- Einrichtung der Präsentationsschicht (Surface, Swapchain, Image Views, Framebuffer)
- Import von Geometriedaten aus separaten Dateien
- Darstellung mehrerer Objekte in einer 3D-Szene
- einfache Benutzerinteraktion mit Kamera und/oder Objekten (z. B. über Tastatur)
- Texturierung von mindestens einem Objekt unter Verwendung einer Mipmap
- Verwendung des Depth-Buffers
- gleichzeitige Verwendung von mindestens zwei verschiedenen Shader-Programmen in einer Szene

### 2.3 Weitere Anforderungen

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
- Strukturierung der Szene durch einen einfachen Szenengraph
- Zerstörung aller Vulkan-Objekte beim Beenden des Programms
- effiziente Verwendung des Gerätespeichers (Staging, Memory Pools)
- feingranulare Synchronisation der Vulkan-Befehle
- gleichzeitige Verwendung von mindestens zwei Texturen in einem Objekt
- Aktivierung von Multisample Anti-Aliasing (MSAA)
- Verwendung einer Umgebungstextur
- Spiegelung der Umgebungstextur auf einigen Objekten
- Verwendung des Stencil-Buffers für die Darstellung eines Spiegels oder eines Portals
- Nebel in einer komplexen 3D-Szene
- Verwendung von Multipass-Rendering (z. B. Schattenwurf von mindestens einer Lichtquelle)
- Berechnung von Szenendaten mit einem Compute-Shader (z. B. für ein Partikelsystem)
- Verwendung des Tessellation- und/oder Geometry-Shaders
- Umsetzung einer weiteren Technik, die in der Vorlesung nicht behandelt wurde

Einige der oben genannten Techniken wurden in der Vorlesung nicht behandelt. Ihr müsst euch also selbstständig in die entsprechenden Techniken einarbeiten. Dies wird für eine besonders gute Note vorausgesetzt.