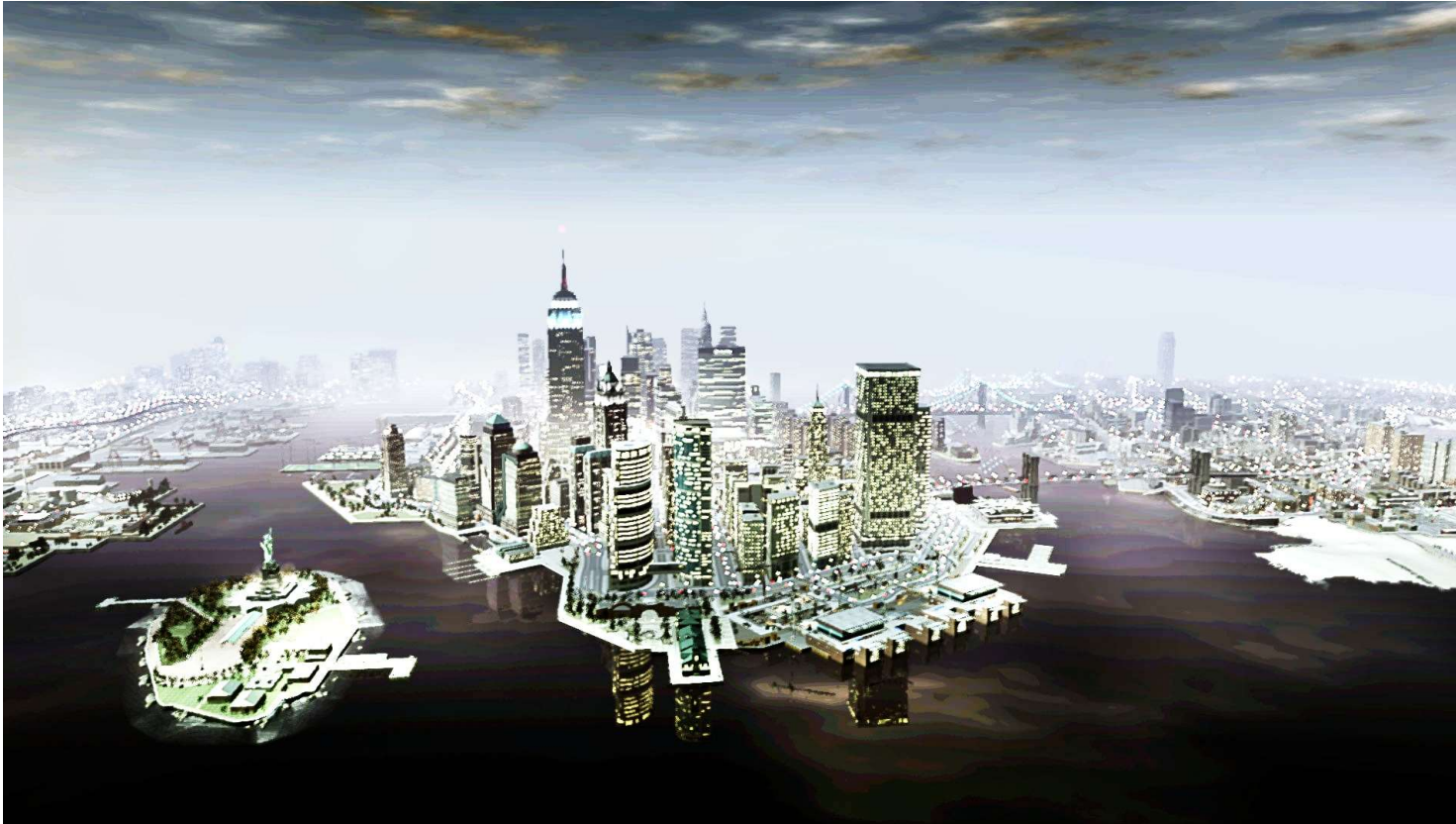


# Computergrafik



Hausarbeit

*Philipp Lensing*

# Prüfungsleistung: Organisatorisches

- Abgabe des Projekts (via Email+Netcase-Link) bis zum **06.09.22** jeweils bis **10:00 Uhr**
- Präsentationen am **08.09.22** Uhr von **8:00 – 20:00 Uhr**
- Präsentationen finden im Raum **SI0036**.
- Konkrete Zeiten für die Präsentationen werden am 06.09.22 per Email mitgeteilt.
- Ablauf Präsentation:
  - 15 Minuten Vortrag
  - 15 Minuten Fragen & Antworten
- **Arbeit in Einzel- oder in Zweier-Gruppen.** Die Bewertung berücksichtigt die Gruppengröße.

# Abgabe

- Abgegeben wird ein **ZIP-Archiv** mit dem Namen „**CG\_SS22\_<Teilnehmername1>\_<Teilnehmername2>.zip**“ mit folgenden Ordnern/Dateien:
  - Ordner „**project**“, Inhalt: komplettes Visual-Studio-, Makefile- oder Xcode-Projekt ohne \*.obj-Dateien mit allen Sourcen. Stellen Sie beim VS-Projekt sicher, dass die Ordner „.vs“, „debug“ und „release“ gelöscht sind, um das Projekt klein zu halten.
  - Ordner „**bin**“, Inhalt: kompilierte Version des ausführbaren Spiels.
  - Datei „**readme.txt**“, Inhalt: Projektteilnehmer, etwaige Beschreibungen zur Steuerung, Angabe von Referenzen & Fremdsourcen, Hinweise zum Quelltext o. ä.
- Die Abgabe erfolgt über **Email** an [p.lensing@hs-osnabrueck.de](mailto:p.lensing@hs-osnabrueck.de) mit dem Titel „**Abgabe CG SS22 <Teilnehmername1>, <Teilnehmername2>**“. Die Email enthält einen entsprechenden **Netcase-Link** zur ZIP-Datei, der mind. 6 Monate gültig ist (keine Email-Anhänge!).

# Prüfungsleistung

Die Hausarbeit besteht aus

- Der Entwicklung eines OpenGL-Programms in C/C++ und
- der Präsentation des Programms mit anschließender Fragerunde.

# Präsentation

- Die Präsentation besteht aus einem
  - 15-minütigen Vortrag und einem
  - 15-minütigen Teil zur Beantwortung von Fragen.
- Publikum sind nur der Prüfer und ggf. dessen Beisitzer.
- Inhalt des Vortrags:
  - **Vorstellung & Erläuterung** der Funktionsweise Ihres Projekts.
  - **Heraushebung technischer Details**, die Sie für interessant bzw. nennenswert erachten.
  - **Kritische Diskussion der Stärken & Schwächen** des Projekts sowie eine **Hervorhebung der besonderen Herausforderungen**.
  - TIPP: Verzichten Sie, wenn möglich, auf die Angabe von Quelltexten, wenn Bilder bzw. Diagramme den Sachverhalt ebenfalls erklären können.
- Die Gesamtnote erhalten Sie nicht unmittelbar nach der Präsentation.

# Präsentation

- Fragen & Antworten
  - Die Fragen beziehen sich auf Inhalte aus der Vorlesung, die im Kontext Ihres Projekts stehen.
  - Die Detailtiefe der Fragen leitet sich aus den Vorlesungsinhalten ab.
  - Ziel der Befragung besteht darin, herauszufinden, ob Sie ein tiefergehendes Verständnis zur Computergrafik entwickelt haben.

# Benotung

- Folgende Kriterien werden bei der Benotung berücksichtigt:
  - Projekt:
    - Funktionalität: Funktionsumfang, Qualität und Komplexität,
    - Fehlerfreiheit,
    - Dokumentation bzw. Lesbarkeit des Quelltextes,
    - objektorientierte Struktur,
    - Laufzeit-Effizienz,
    - Benutzerfreundlichkeit,
    - Gruppengröße.
  - Präsentation:
    - Vortrag:
      - Verständlichkeit & Fokus
      - Zeitmanagement
    - Befragung:
      - Korrekte & prägnante Beantwortung der Fragen

# Projekt: 3D-Spiel

- Entwicklung in C/C++.
- Nutzung der Klassen und Funktionen aus dem Praktikum ist erlaubt.
- Es dürfen keine externen Bibliotheken zur linearen Algebra & Darstellung genutzt werden außer OpenGL, GLEW, GLFW & GLUT.
- Quelltextübernahmen sind erlaubt, wenn deutlich kenntlich gemacht wird, wer der eigentliche Urheber des Quelltextes ist.
  - Kommentare in Quelltext & readme.pdf.
  - Übernahmen ohne Kenntlichmachung werden als Täuschungsversuch angesehen und führen zu einer nicht bestandenen Prüfungsleistung.
- Readme.pdf:
  - Kurze Bedienungsanleitung
  - Fremde Quellen
  - Besonderer Fokus auf bestimmte Funktionen oder elegante Lösungen o. ä.
  - Weitere Anmerkungen



# 3D-Spiel

- Bis zum 24.06.22 (letzter Vorlesungstermin) Spielidee mitteilen.
- Das Spiel muss die auf der folgenden Folie beschriebenen Voraussetzungen erfüllen.
- Wird das Spiel anhand eines Online-Tutorials entwickelt, so muss das genutzte Tutorial zwingend angegeben werden. Das Spiel muss **wesentliche** Unterschiede zum Tutorial aufweisen!

# 3D-Spiel

## Voraussetzungen:

- Die genutzten bzw. erzeugten 3D-Modelle besitzen Materialeigenschaften & Texturen.
- Objektmaterialien in Form von Phong-Reflexionskoeffizienten & Texturen werden korrekt ausgelesen und dargestellt.
- Die Darstellung der Modelle erfolgt ausschließlich über Shader.
- Die Geometriedaten der Modelle werden effizient in Vertex- und Index-Buffern auf der Grafikkarte gespeichert.
- Die Kamera ist dynamisch (kann durch Eingabe gesteuert werden oder bewegt sich mit Objekten).

# 3D-Spiel: Beispiel



# Vorlesungsinhalte

## 1. Bildsynthese, Whitted Raytracing & 3D-Repräsentationsschemata

- Der Raytracing-Algorithmus.
- Farben im RGB-Raum & Farbwahrnehmung.
- Constructive Solid Geometry, Voxel- & Polygonmodelle.
- Grundlagen der Vektorrechnung
- Einfacher Strahlen-Dreieck-Kollisionstest.

# Vorlesungsinhalte

## 2. Beleuchtungsmodelle

- Lokale & globale Beleuchtungsmodelle
- Optische Phänomene
- Phong-Reflexionsmodell
- Analytische Lichtquellenrepräsentationen
- Reflexion & Transmission von Lichtstrahlen
- Dispersion

# Vorlesungsinhalte

## 3. Echtzeit-Bildsynthese:

- Framerate/Frametime
- Simulation-Loop
- Frameunabhängige Animationssteuerung
- Double-Buffering, V-Sync.
- CPU- & GPU-Workbalancing
- OpenGL/GLUT
- Rendering-Pipeline

# Vorlesungsinhalte

## 4. Transformations-Pipeline:

- Objekt-, Welt-, Kamera, normalisierter Bild- und Fensterraum
- Affine & projektive Transformationen
  - Translation
  - Rotation
  - Skalierung
  - Orthogonale und perspektivische Projektion
  - Szenegraphen
  - Kamera-Transformationen

# Vorlesungsinhalte

## 5. Clipping, Culling & Rasterung:

- Back-Face-Culling
- Clipping von Linien: Cohen-Sutherland-Algorithmus
- Clipping von Polygonen: Sutherland-Hodgman-Algorithmus
- Rasterung von Liniensegmenten: Bresenham-Algorithmus
- Rasterung von Polygonen: Scanline-Algorithmus



# Vorlesungsinhalte

## 6. Rasterung & Texturen:

- Perspektivisch korrigierte Interpolation von Vertexattributen bei Rasterung.
- Glättung von Normalen.
- Flat-, Gouraud- & Phong-Shading
- Texture-Mapping
  - Erzeugung von Texturkoordinaten
  - Bildbasierte & prozedurale Texturen
  - Textur-Adressierungsarten
  - Arten von Texturen: Diffuse-, Specular-, Emissive-, Bump-, Normal-, Displacement- & Environment-Maps
  - Filterung von Texturen: Nearest-Neighbor, bilinear, Mip-Mapping, trilinear, anisotrop

# Vorlesungsinhalte

## 7. Shader-Programmierung

- Shader-Sprachen & -Modelle
- Unified Shader-Model
- Vertex- und Fragment-Shader mit GLSL
- GLSL Datenmodell & Sprachspezifikationen
- Shader-Beispiele:
  - Rimm-Lights & Outlines
  - Phong-Shader
  - Toon-Shader

# Vorlesungsinhalte

## 8. Postprocessing

- Raster-Operations mit OpenGL
- Offscreen-Rendering mit OpenGL
- Beispiel-Implementierungen
  - Weichzeichnen (Box/Gauß)
  - Farben invertieren
  - Graustufen erzeugen
  - Gamma-Korrektur
  - Kanten hervorheben (Sobel-Operator)
  - Bloom-/Glow-Effekt
  - Depth-of-Field-Effekt
  - Konturzeichnung für Toon-Shading
- Screen-Space-Ambient-Occlusion
- Deferred-Shading

# Vorlesungsinhalte

## 9. Szenenmanagement

- Frustum-Culling
- Bounding-Volumes
- Bounding-Volume-Hierarchies (Hüllkörper-Hierarchien)
  - Octree, Quadtree, BSP-Tree
- Portal-Systeme
- Potentially Visible Sets
- Occlusion-Culling & Hardware-Occlusion-Queries
- Diskrete Level-of-Detail-Ansätze
- Progressive Meshes

Vielen Dank für Ihre Aufmerksamkeit  
& viel Erfolg!