



Computergrafik II - Einführung -

Bachelor Medieninformatik
Wintersemester 2011

Prof. Dr.-ing. Hartmut Schirmacher
<http://schirmacher.beuth-hochschule.de>
hschirmacher@beuth-hochschule.de



Gliederung



- Computergrafik II Motivation
 - Lernziele
 - Anwendungsbeispiele
 - Was kommt nach CG2?
- Kennenlernen
- Kursüberblick
- Organisatorisches
- Die erste Übungsaufgabe





Computergrafik II Motivation



3

Warum Computergrafik II?



- Computergrafik I: Grundlagen
 - Fundamente der graphischen Datenverarbeitung (Rasterisierung, Geometrie, Transformation, Projektion, Farbe, Beleuchtung, ...)
- Computergrafik II: Bildsynthese (Rendering)
 - Wie funktioniert fotorealistische Bilderzeugung?
 - Welche grundlegenden Ansätze gibt es?
 - Welche Konzepte und Algorithmen benötige ich, um selbst Bilder von 3D-Szenen zu erzeugen?
- Wenn Sie denken, daß Sie das schon wissen...
 - ... schauen Sie die nachfolgenden Beispiele an.

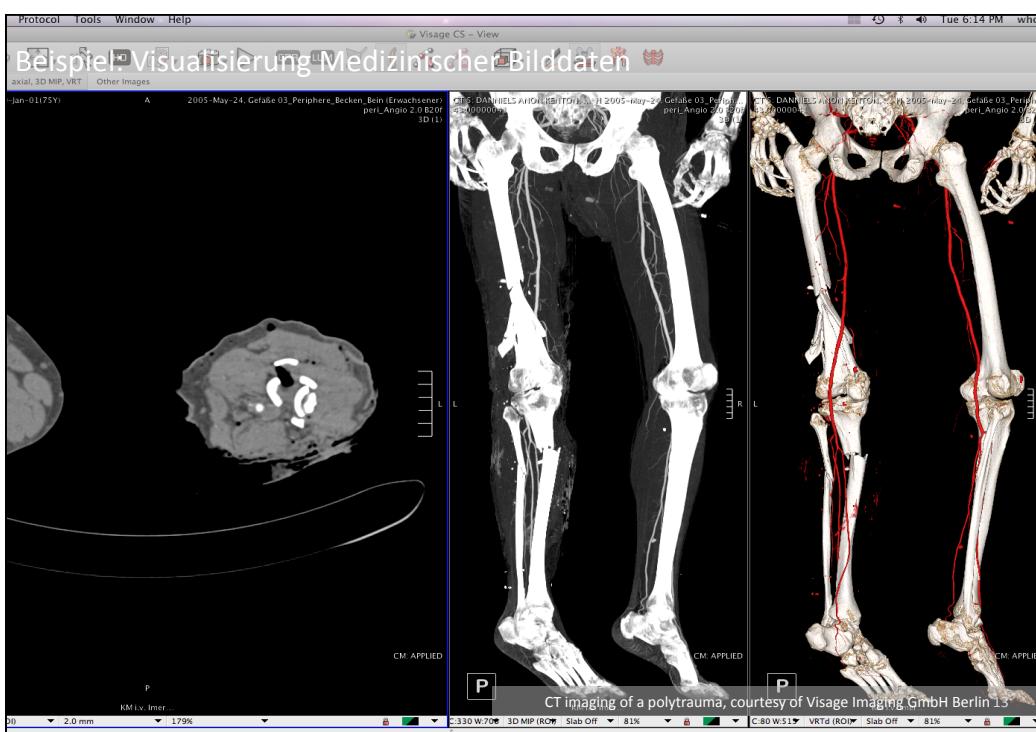
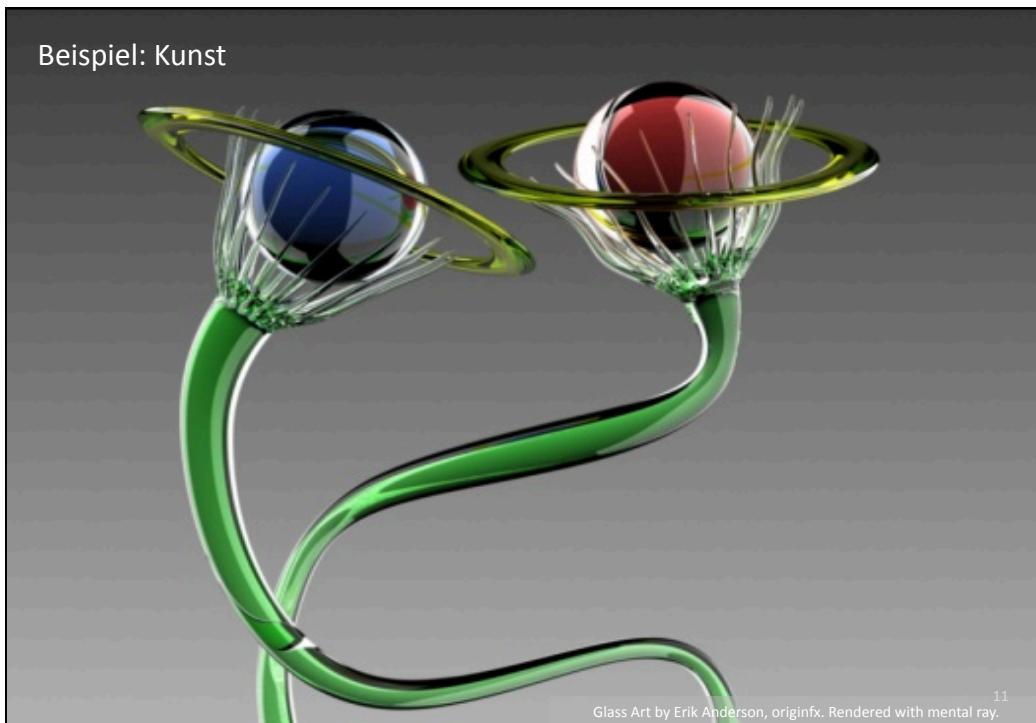


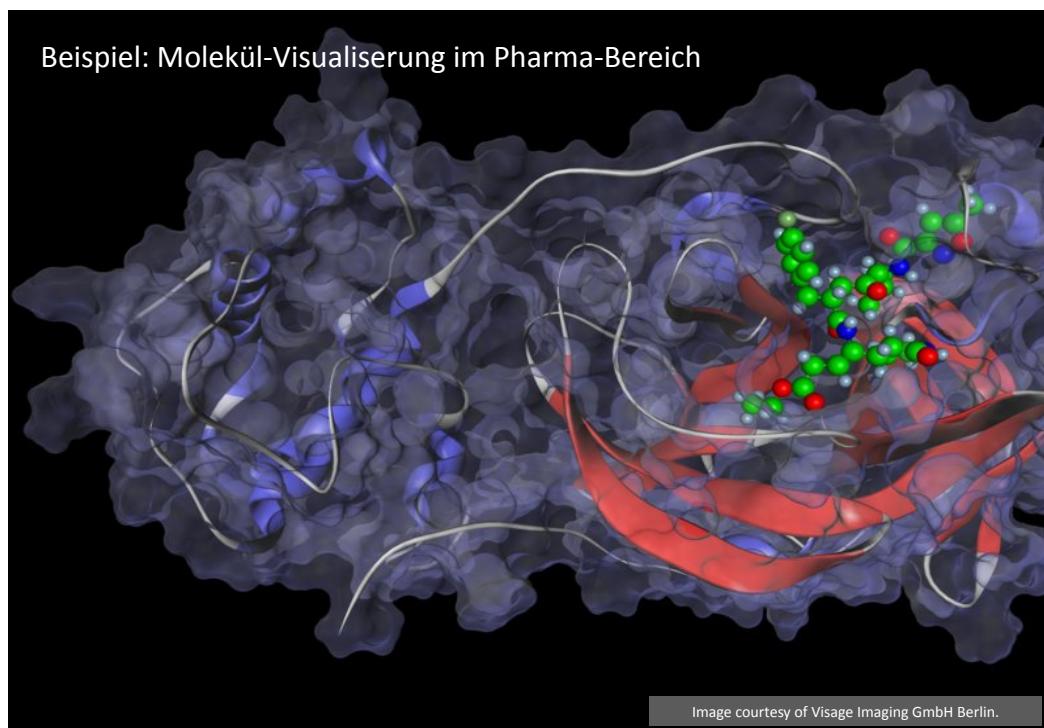
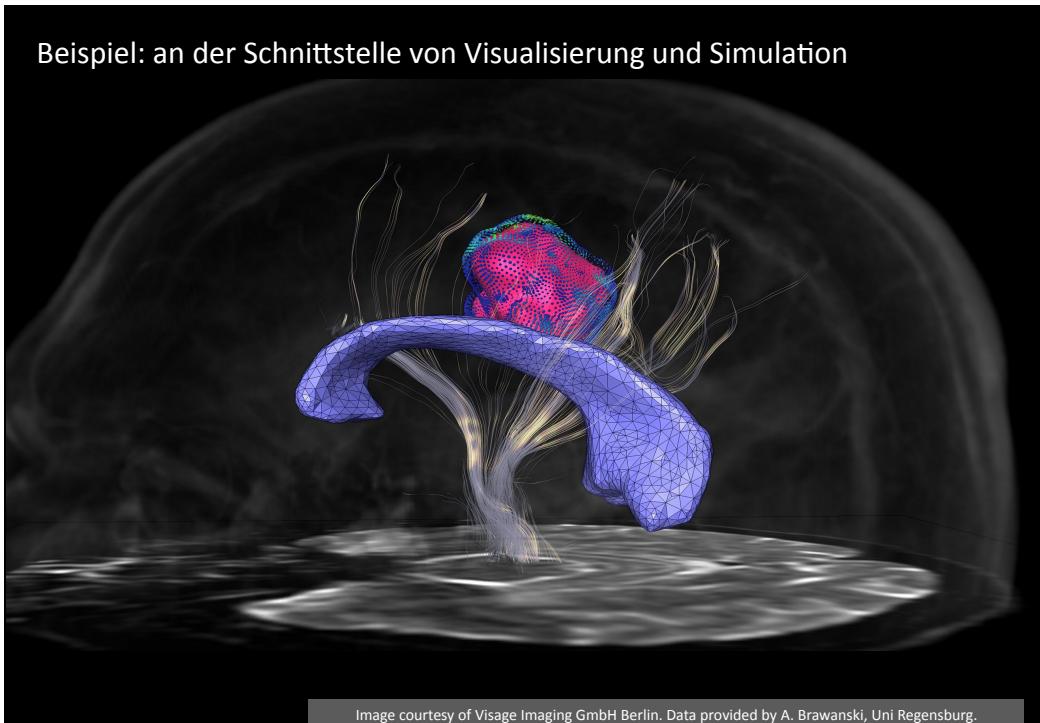
2











Anwendungsgebiete



- Anwendungsgebiete der Computergrafik
 - Interaktive Computerspiele
 - Film, Zeichentrick, Special Effects
 - Werbung
 - Architektur
 - CAD/CAM: Computer Aided Design + Manufacturing
 - Virtual Reality und Augmented Reality
 - Simulation
 - Bio-medizinische Visualisierung
 - Informationsvisualisation
 - Mensch-Maschine-Kommunikation (HCI)
 - Kunst



Kursziele



- Lern- und Leistungsziele dieses Kurses
 - Sie erlernen die Grundlagen der 3D-Bilderzeugung
 - Sie programmieren beispielhaft die beiden grundlegenden Ansätze zur fotorealistischen Bildsynthese (Raytracing + GPU-Shader)
 - Sie erwerben ein Gefühl für Komplexität, Performanz und Fallstricke bei den verschiedenen Rendering-Algorithmen
 - Sie lernen, je nach Fragestellung den richtigen 3D-Rendering-Ansatz auszuwählen



Was kommt nach Computergrafik II?



- Programmier-Projekte
 - Nach dieser Grundausbildung sind Software-Projekte die beste Möglichkeit, Computergrafik zu vertiefen
- Weiterführende Kurse im (künftigen) Master-Programm
 - Real Time Rendering
 - Game Engine Architecture
 - Interactive Systems



Kennenlernen



Hartmut Schirmacher - Lebenslauf



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

10/2011 – ...	Professor für Computergrafik und Programmierung Beuth Hochschule für Technik Berlin
2004 – 2001	Senior Product Manager, Director Product Management Mercury Computer Systems / Visage Imaging,
2003	Promotion zum Dr.-ing
2002 – 2004	Projektmanager Virtual Reality Solutions & Produktmanager amira® Indeed – Visual Concepts GmbH, Berlin
1999 – 2002	Wissenschaftlicher Assistent Max-Planck-Institut für Informatik, Saarbrücken
1998 – 1999	Wissenschaftlicher Assistent Universität Erlangen-Nürnberg
1998	Zuse Institut Berlin (6 Monate)
1996	iMAGIS, Grenoble (6 Monate)
1992 – 1998	Dipl.-Inf., Universität Erlangen-Nürnberg

Hartmut Schirmacher – Lebenslauf



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

- (Frühere) Forschungsgebiete
 - Globale Beleuchtung (Radiosity, Volume Radiosity)
 - Real Time Rendering (Hybridmethoden)
 - Image Based Rendering (Lichtfelder)
 - Bildbasierte Rekonstruktion von Oberflächen
 - Bildanalyse und Visualisierung (Biomedizin)
- Industrie-Erfahrung
 - Virtual-Reality-Lösungen (Hardware + Software)
 - Bildrekonstruktion für CT, MR, Elektronenmikroskope, ...
 - Medizinische und wissenschaftl. Visualisierung
 - Bilddaten-Management
 - Radiologie als Anwendungsdomäne

Hartmut Schirmacher – Persönliches



- Seit neun Jahren in Berlin
 - Ursprünglich aus Gummersbach im Bergischen Land (VfL, Heiner Brandt, Hella von Sinnen, ...)
 - Steglitz, Kreuzberg, Mitte
- Familie
 - Lebensgefährtin
 - Sohn Paul (12/2010)
- Hobbies
 - Fotografie
 - Beachvolleyball
 - Chor (www.tonikum.org)



Hartmut Schirmacher – Kontakt und Infos



- Webseite (im Aufbau...)
 - <http://schirmacher.beuth-hochschule.de>
 - Kursmaterial + Neuigkeiten
 - Bitte die FB VI **Lehrkraftnews** verwenden!
- Email
 - hschirmacher@beuth-hochschule.de
- Büro und Sprechzeiten
 - Haus Gauß, Raum 122
 - Mittwochs 14:00 – 15:30
 - Wenn möglich bitte vorher per Email ankündigen
 - Wenn nötig, auch Treffen zu anderer Zeit vereinbar!



Ihr Dozent und Sie



- Ich bin letztendlich *für Sie* hier
 - Meine Arbeit dreht sich um Ihre Ausbildung und Karriere... arbeiten Sie aktiv mit mir zusammen!
 - Ich bin neu. Ich mache Fehler. Bitte nicht übelnehmen, sondern Feedback geben und helfen
 - Ich bin offen für Kritik und Anregungen!
 - Kritik wirkt sich nicht auf die Note aus!
 - „Ich verstehe X nicht“ beeinflusst **nicht** die Note!
 - Wer (konstruktiv) kritisiert oder kommentiert, ist aktiv und beschäftigt sich mit der Materie → **Verbesserung** der Wahrnehmung!!!



Kennenlernen



- Und jetzt sind Sie dran...
- Schnelle Intro-Runde
 - Ihr Name und Semester
 - Frühere / aktuelle Studiums- oder Berufserfahrung
 - Was möchten Sie mit dem Bachelor erreichen, was möchten Sie danach *idealerweise* machen?
 - Was Sie uns sonst noch (kurz) mitteilen möchten ...





BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

Kursüberblick



28

Computergrafik



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

- Allgemeine Definition
 - **any use of computers to create or manipulate images**
(Shirley: *Fundamentals of Computer Graphics*)
- Untergebiete
 - Modellierung
 - **Rendering oder Bildsynthese**
 - Animation
 - (Interaktion)
- Verwandtes Gebiet:
 - Bildanalyse / Bildverstehen (Computer Vision)

Fokus dieses Kurses



29

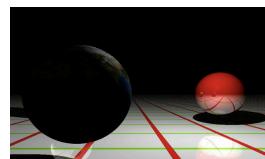
Zwei Grundlegende Ansätze zur Bildsynthese



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

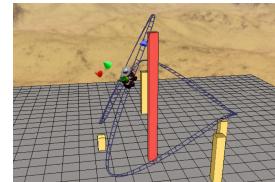
■ Ray-Tracing-Ansatz

- Sende einen „Kamera-Strahl“ oder „Primärstrahl“ durch jeden Pixel des Bildes in die Szene hinein
- Für jeden Treffer sende rekursiv „Sekundärstrahlen“ in Richtung anderer Objekte und Lichtquellen
- Sammle einfallendes Licht und berechne Farbe des Treffer-Punktes



■ Rasterisierungs-Ansatz

- Berechne die Beleuchtung für jeden Vertex meiner geometrischen Objekte
- Zerlege die Geometrie in Fragmente und projiziere sie auf das Bild
- Berechne die finale Farbe für jede Fragment und kombiniere die Fragmente zu Pixeln



30

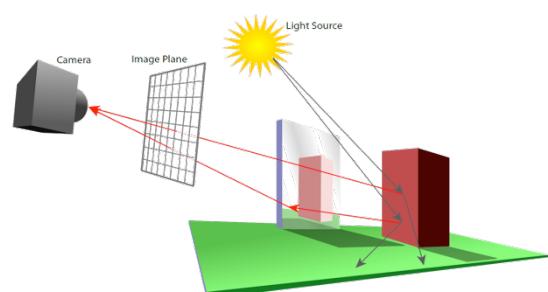
Ray Tracing: Themengebiete



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

■ Themengebiete beim Raytracing

- Der Raytracing-Algorithmus
- Geometrische Repräsentierung von Objekten (Kugel, Ebene, Quader, ...)
- Schnittberechnung von Strahlen mit Objekten
- Beleuchtungsberechnung
- Rekursives Raytracing
 - Reflexion
 - Refraktion
 - Schatten
- Szenengraphen für Raytracing
- Beschleunigungsstrukturen



Ray Tracing: Praktische Relevanz



- Raytracing in der Industrie:
 - Höchste Bildqualität, z.B. für Werbung / Film
 - Architekturmodelle mit realistischer Beleuchtung
 - Interaktive Exploration riesiger Polygonmengen

- Aktuelle Trends beim Raytracing:
 - Traditionell eher im „Offline“-Betrieb und auf der CPU
 - Zunehmend auch für interaktive Visualisierung
 - Zunehmend auch auf der GPU



Raytracing: Auf echter Physik basierende Berechnung



Interaktives Raytracing für riesige Modelle

BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

- Boeing 777: ~350 Millionen individuelle Polygone,
~30 GB Speicherplatz

Image courtesy of Philipp Slusallek, Universität des Saarlandes.

Interaktives Raytracing für riesige Modelle

BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

- 90×10^{12} (Billionen) Dreiecke

Images courtesy of Philipp Slusallek, Universität des Saarlandes.

Raytracing auf der GPU mittels OptiX

BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

Images from OptiX demos on nvidia.com

36

Bildsynthese mittelse Rasterisierungs und Shadern

- Themengebiete beim Rasterisierungs-Ansatz:
 - Allgemeines Modell der Rasterisierungs-Pipeline
 - Einführung in das OpenGL ES 2.0 API
 - Im wesentlichen die Kernfunktionen von OpenGL 4
 - Auch via WebGL und auf mobilen Geräten verfügbar
 - Shaderprogrammierung
 - Vertex-Shader
 - Fragment-Shader
 - Andere Shader-Typen
 - Nochmal Szenengraphen
 - Traversierung von Szenengraphen
 - Designaspekte für Szenengraphen

Model-View Transformation
Per-Vertex Lighting
Projection
Clipping
Rasterization and Interpolation
Visibility Resolution

36

Praktische Relevanz des Rasterisierungs-Ansatzes



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

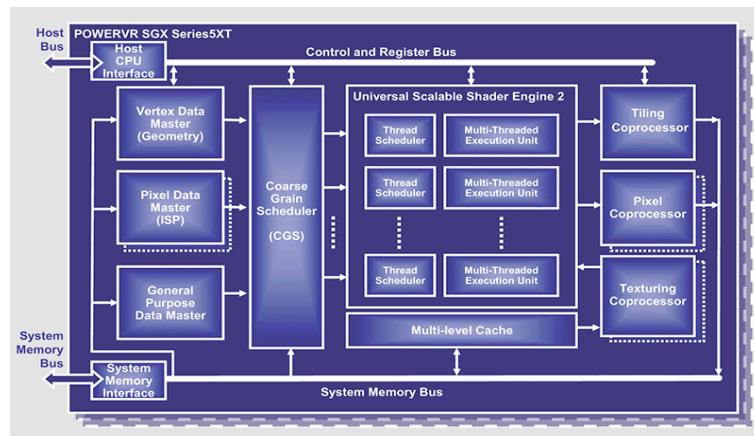
- Rasterisierung / GPUs in der Industrie:
 - Fast alle interaktiven Applikationen (und speziell Spiele) basieren auf Rasterisierung
 - Moderne Generationen von GPUs machen diesen Ansatz sehr performant, sogar auf mobilen Geräten
 - Im wesentlichen treibt die Spieleindustrie diese rasante Entwicklung weiter voran
- Trends bei der Rasterisierung / den GPUs:
 - Immer komplexere Rendering-Effekte
 - Eine immer flexiblere Rasterisierungs-Pipeline – z.B. für allgemeine Berechnungen (general purpose GPU computing, GPGPU)



Beispiel: PowerVR (früher VideoLogic)

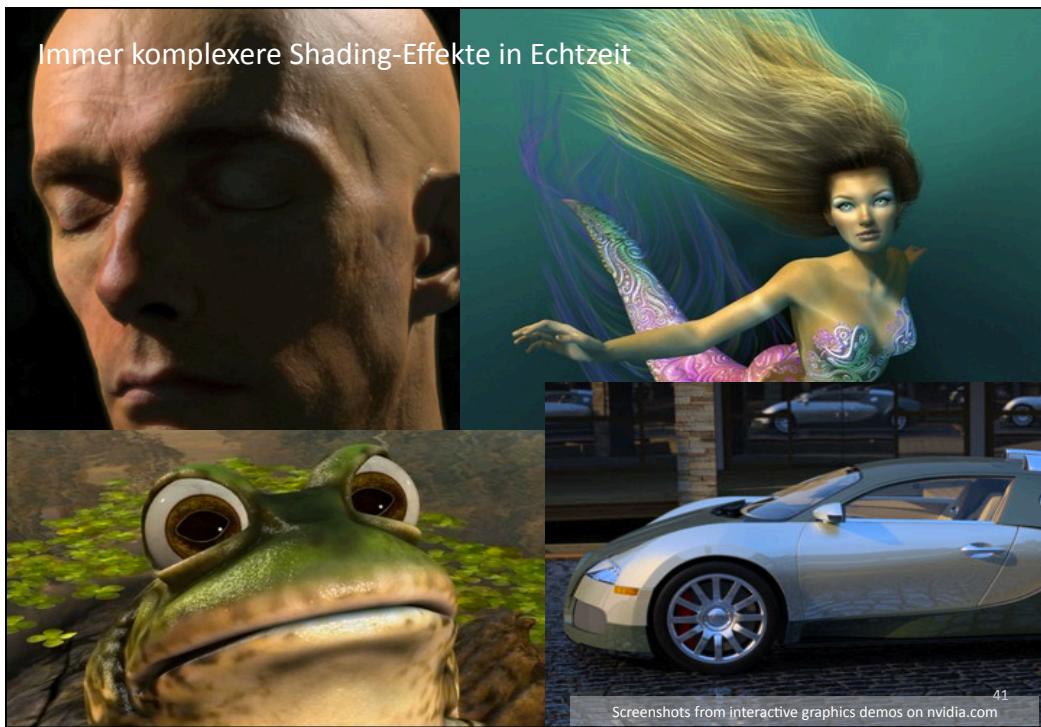


BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences



- GPU-Design (System-On-A-Chip) von vielen Herstellern lizenziert, unterstützt OpenGL ES 2.0
- Zum Beispiel in [iPhone 4](#), [iPad](#), [iPod touch](#), [Apple TV](#), [Nokia N900](#), [Sony Ericsson Vivaz](#), [Motorola Droid/Milestone](#), [Archos 70](#), [Palm Pre](#), [Galaxy S](#), [Galaxy Tab](#), [Samsung Wave S8500](#), [Samsung Wave II S8530](#), [Samsung Galaxy SL](#)







Organisatorisches



42

Vorlesung und Folien



- Folien auf Homepage verfügbar
 - Hoffentlich am Samstag vor der jeweiligen Vorlesung
- Englisch-Deutsch
 - Sämtliche ernstzunehmende Fachliteratur ist Englisch
 - Ich werde des öfteren die englischen Begriffe verwenden, oder deutsche und englische Begriffe synonym
 - Überstellungsfragen am besten direkt fragen!!!
- Folien basieren stark auf denen von Prof. Tramberend
 - Wir sprechen unsere Inhalte ab und entwickeln sie gemeinsam weiter
- Feedback zur Vorlesung und den Folien erbeten!



42

Zeitaufwand und Leistungsbewertung



- Verbindliche Festlegung des Zeitaufwands im *Modulhandbuch Medieninformatik Bachelor*

Seminar. Unterricht (SU)	36 h
Übungen	36 h
Selbstlernzeit	72 h

{}

2 h Übung + 4 h Selbstl. = 6 h / Woche

- Gesamtnote

- Bewertung 1.0, 1.3, 1.7, 2.0, 2.3, ..., 5.0
- 2/3 Übungsnote + 1/3 Klausurnote
- Übungsnote: nach Bearbeitungszeit gewichtetes Mittel
- Aktive Teilnahme am SU kann nur helfen



Klausur



- Teilnahmebedingung
 - Erfolgreiche Abgabe **aller** Übungsaufgaben
- Zeitpunkt
 - Vorletzte Woche der Vorlesungszeit
 - Nachschreibetermin Anfang des folgenden Semesters
- Format
 - Multiple Choice
 - **Keine** Hilfsmittel
- Orientierungshilfe (unverbindlich!)
 - Vorangegangene Klausuren von Prof. Tramberend



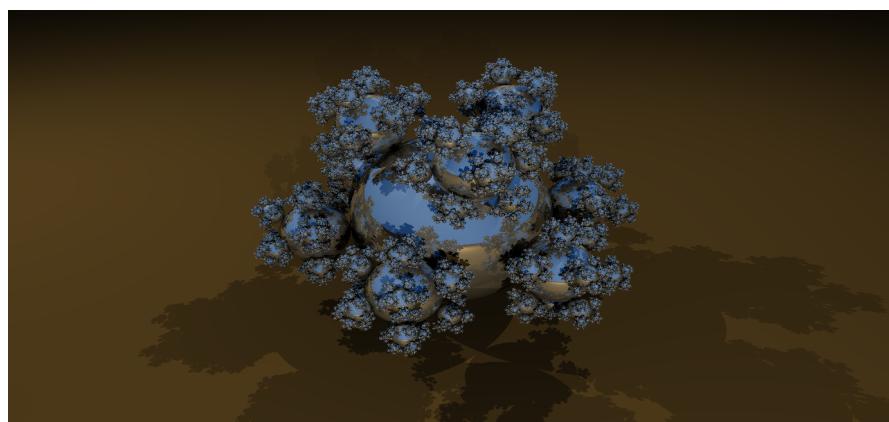
Übungsaufgaben

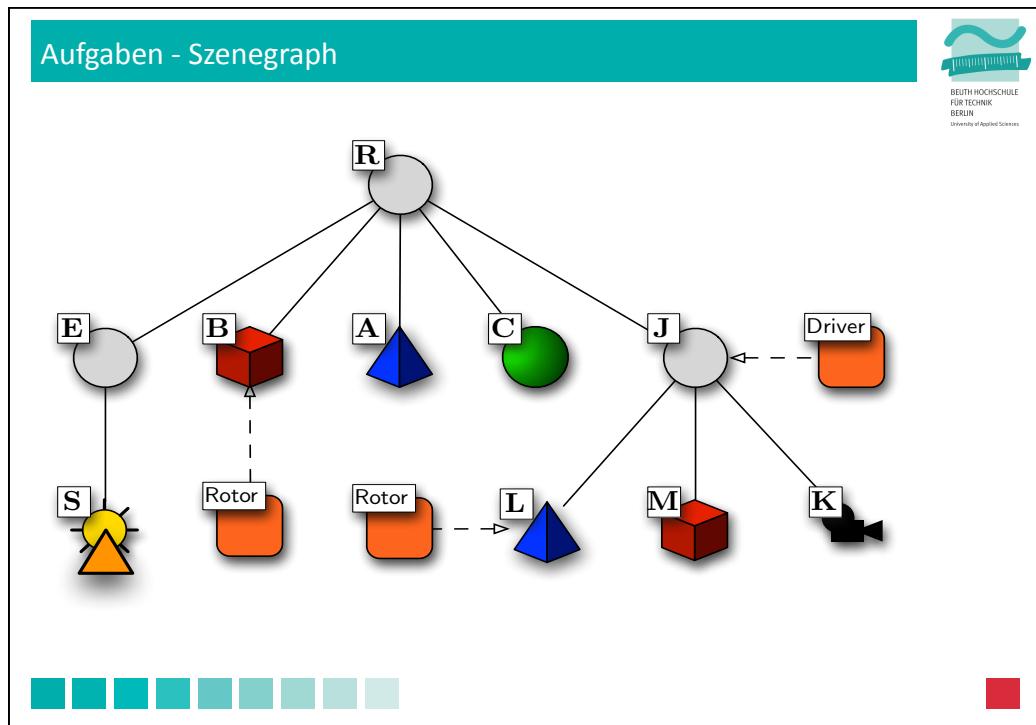
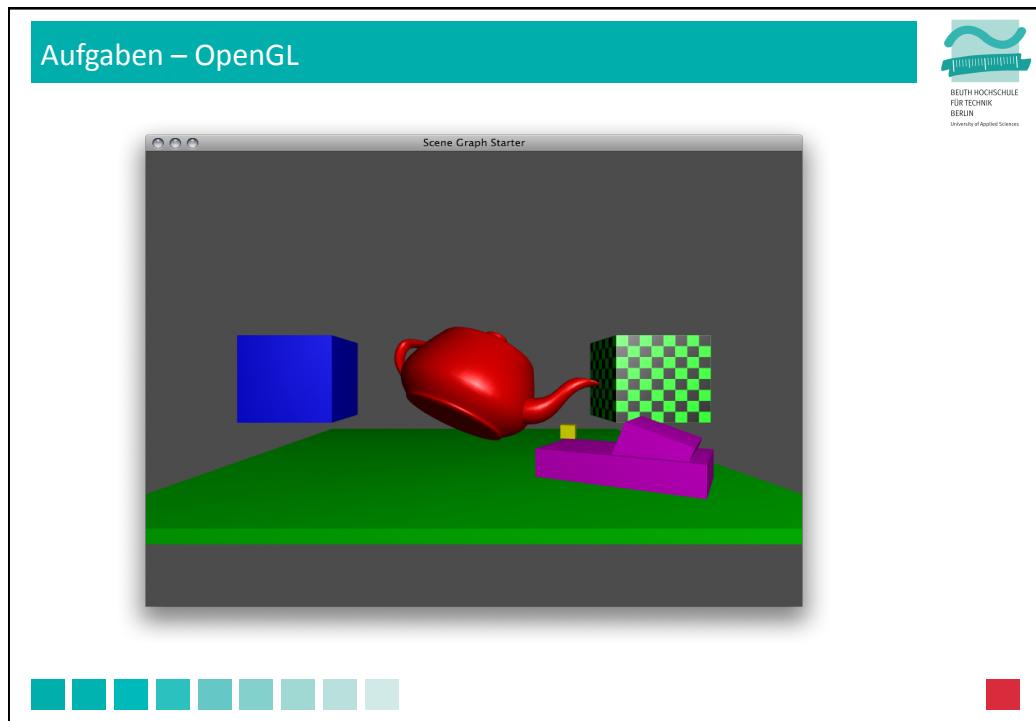


- Mehrere Aufgabenstellungen bis Semesterende
- Aufgaben bauen teilweise aufeinander auf
- Bearbeitung in 2-er Gruppen
- Bearbeitung in den Übungen und in Selbstlernzeit
- Abgabe zu festen Terminen
- Aufgaben größtenteils benotet
- Notengewicht proportional zur Bearbeitungszeit



Übungsaufgaben – Ray Tracing





Übungsaufgaben – Infrastruktur



- Implementierungsplattform
 - Beliebig
 - Laborrechner laufen unter Windows
 - Eigenes Laptop, iPad, iPhone kein Problem
 - Späteren Aufgaben benötigen OpenGL ES 2.0
- Programmiersprachen
 - Empfohlene und unterstützte Sprachen: Java und C++
 - Andere möglich (C#, Javascript, Ruby, ...)
 - Programmiersprachen-Hilfe kann nur in begrenztem Umfang gewährt werden... Es wäre gut, wenn Sie die Sprache grundlegend beherrschen.



Übungsaufgaben – Abgabe



- Abgabe per Email
 - Ein komprimiertes Archiv pro Gruppe
 - Programmtext, Bilder, Folien, Filme... **KEINE Compilate**
 - Als .zip oder .tar.gz (kein .rar oder .7z)
 - Dateiname cg2-aN-Nachname1-Nachname2-zip
 - Beispiel: [cg2-a1-müller-schmidt.zip](#)
- Abgabetermin
 - Jede Aufgabe hat einen festen Termin
 - Verspätete Abgaben bis zu zwei Wochen in der Regel **möglich**
 - **Punktabzug** 2/3 Note je angefangener Woche Verspätung



Fremde Lösungsanteile und Quellenangaben



- Fremde Lösungsanteile
 - In vernünftigem Maße erlaubt – nicht jedoch für die Hauptsache der Aufgabe
 - Eigene Lösungen werden höher bewertet
- Quellennachweis
 - Fremde Anteile müssen im Code / in der Doku **gekennzeichnet** sein
 - Verwendung von Fremdanteilen muß bei der Abnahme der Aufgabe in der Übung **mündlich erwähnt** werden
- Fremde Anteile ohne Quellenangabe = **Note 5.0**



Plagiate



- Abgabe eines Plagiats
 - Fremde Lösung oder wesentliche Teile davon
 - Ernsthafter Betrugsversuch
 - Abgabe wird als nicht bearbeitet abgewiesen
 - Für Plagiatoren **und** ggf. Autoren der Fremdlösung
 - Konsequenzen
 - Übungsaufgabe **nicht bestanden**
 - **Keine Zulassung** zur Klausur
 - Bei schweren / dreisten Fällen **Meldung im Dekanat**
 - Kann zu **Exmatrikulation** führen
- Wenn das Plagiat erst später entlarvt wird...
 - ... wird die vorher zugeteilte Bewertung nachträglich aberkannt, und die o.a. Konsequenzen drohen.





Allg. Fragen zu
Vorlesung & Übung?



54



Die erste Aufgabe



55

Ray Tracing Warm Up

BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Prof. Dr.-Ing. Hartmut Schirmacher
Fachbereich Informatik und Medien

Computergrafik 2 / Aufgabe 1: Warmup

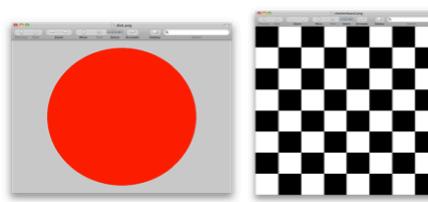


Abbildung: Erzeugen Sie Pixel für Pixel ein 8x8-Schachbrettmuster und eine 90%-Kreisscheibe.

In den kommenden Wochen werden Sie einen einfachen Raytracer implementieren. Dieser soll die erzeugten Bilder im PNG-Format auf der Festplatte ablegen können. Die erste Aufgabe besteht aus einigen vorbereitenden Arbeiten zur Eingewöhnung in die Entwicklungswerzeuge und die Arbeitsweise des Raytracers.

