

25 Graphical Visualisation Technologies	
Graphical Visualization Technologies	
Semester	Wahlpflichtbereich
Credit Points	5
Pflicht/ Wahlpflicht	Wahlpflicht
Häufigkeit des Angebotes/ Verwendbarkeit	Jedes Semester nach Bedarf der Hochschulen im VFH-Verbund
Modulverantwortliche(r)	Prof. Dr. Felix Gers, Beuth Hochschule für Technik Berlin
Lerngebiet	Medieninformatik
Teilnahmevoraussetzungen	empfohlen: Kenntnisse in HTML und JavaScript
Lernergebnisse	<p>Die Studierenden erlernen Graphische Algorithmen und Renderingverfahren einzusetzen und damit Grafikanwendungen für das Internet zu entwickeln. Sie können die Möglichkeiten und Grenzen der vermittelten Techniken einschätzen und diese praktisch anwenden. Nach dem erfolgreichen Abschluss des Studienmoduls, sind die Studierenden in der Lage:</p> <ul style="list-style-type: none"> • Die 3D-Rendering-Pipeline nachzuvollziehen • JavaScript und WebGL im Rahmen der gestellten Programmieraufgaben zu verwenden • Befehle der WebGL-API anzuwenden • Den Aufbau polygonaler 3D-Modelle zu verstehen • 3D-Grundkörper algorithmisch zu erzeugen • 3D-Szenen zu erstellen und zu beleuchten • Modelle zu texturieren • Daten zu laden in interaktiv zu visualisieren • Browseranwendungen mit 2D und 3D Echtzeitgrafik unter Verwendung von WebGL (OpenGL) und der GLSL (OpenGL Shading Language) zu gestalten • Shader-Programmierung auf der GPU (Graphics Processing Unit) praktisch anzuwenden • Im allgemeinen Hardware-gestützte 3D-Computergrafik unter Einsatz programmierbarer Grafikkbeschleuniger zu nutzen • die Möglichkeiten und Grenzen der vermittelten Techniken einzuschätzen • ein Projekt von der Konzeption bis zur gestalterischen und praktischen Umsetzung zu erstellen
Prüfungsvorleistung	keine

Medien-/ Lernform	Multimedial aufbereitetes Online-Studienmodul zum Selbststudium mit zeitlich parallel laufender Online-Betreuung (E-Mail, Foren, Chat, Webkonferenzen, Einsendeaufgaben u. a.)
Arbeitsaufwand	Selbststudium: 123 h Betreutes Lernen: 27 h
Präsenzinhalte	Praktische Übungen, Besprechung der Einsendeaufgaben, gemeinsame Bearbeitung weitere Aufgaben und Übungen, Klärung inhaltlicher Fragen.
Prüfungsform	Hausarbeit/Projekt
Literatur	Tony Parisi, „Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages“, O'Reilly Kouichi Matsuda; Rodger Lea Matsuda, „WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL“, Addison-Wesley Diego Cantor; Brandon Jones, „WebGL Beginner's Guide“, Packt Publishing. Tomas Akenine-Moller, Eric Haines, Naty Hoffman, „Real-Time Rendering“, Taylor & Francis
Vertiefungsrichtung	Interactive 3D
weitere Hinweise	

Studieninhalte

In diesem Modul werden wir uns auf interaktive 3D-Visualisierungen, das bedeutet auf die Echtzeit-Darstellung von dreidimensionalen Objekten und Szenen, fokussieren. Die Einsatzgebiete von 3D-Visualisierungen sind vielfältig, dazu gehören Filme, Computerspiele, begehbare Gebäude und Fertigungsstätten und allgemeiner die grafische Darstellung von Daten und Prozessen. In den Übungen werden interaktive 3D-Szenen aus Grundkörpern und auf der Basis von Daten erstellt.

Die Visualisierungen sollen auf Webseiten also mit Internet-Browsern aufgerufen und ausgeführt werden können. Daher werden wir zur technischen Umsetzung den Standard WebGL einsetzen.

WebGL ist die Basistechnologie für interaktive 3D-Visualisierung im Internet. Die erlernte Technologie ist mittels OpenGL auf grafischen Anwendungen allgemein übertragbar.

Im Kursmaterial werden die theoretischen und technischen Grundlagen von Echtzeit-3D-Visualisierung erläutert und an Hand von praktischen Beispielen erprobt. Themenbereiche sind: 3D-Rendering-Pipeline, der Aufbau von 3D-Modellen, die Konstruktion von interaktiven 3D-Szenen mit WebGL, Transformationen – Bewegung in der Szene, Kamera und Beleuchtung, Shader-Programmierung mit GLSL (OpenGL Shading Language), Texturierung und das Laden und Darstellen von Daten.

Gliederung des Studienmoduls

1. Einleitung
2. Arbeitsumgebung
3. JavaScript Language

4. Web Graphics Library (WebGL)
5. WebGL Application Interface (API)
6. Geometrie
7. Farbe
8. Kurven und Flächen
9. Grundkörper
10. Kamera und Perspektive
11. Transformationen und Bewegung
12. Shader
13. Shading Language Programmierung
14. Beleuchtung
15. Texturen
16. Daten
17. Navigation
18. Dimensionsreduktion von Daten