



Fakultät Informatik

Studiengang Game Engineering und Visual Computing (Master)

# Modulhandbuch

Stand: SS 2021

Februar 2021

---

Prof. Dr. E. Müller  
Studiendekan der Fakultät Informatik

---

Prof. Dr. B. Dreier  
Studiengangkoordinator

---

Prof. Dr. U. Göhner  
Vorsitzender der Prüfungskommission

## Inhaltsverzeichnis

<b>1 Ziele und Aufbau des Studiengangs Game Engineering und Visual Computing ....</b>	<b>2</b>
<b>2 Begriffserläuterungen .....</b>	<b>6</b>
<b>3 Modulbeschreibungen .....</b>	<b>8</b>
GEM1101 Algorithmen für Real Time Rendering .....	8
GEM1102 Simulation und Datenvisualisierung .....	10
GEM1103 Computer Vision .....	12
GEM1104 Advanced Game Engineering .....	14
GEM1105 Augmented Reality .....	16
GEM2101 Serious Games .....	18
GEM2102 Deep Learning .....	20
GEM2103 Algorithmen und Strategien zur Entscheidungsunterstützung .....	22
GEM2104 Parallelprogrammierung .....	24
GEM2105 Physically Based Rendering .....	26
GEM2106 Softwarearchitektur .....	28
GEM2107 Mobile Roboter .....	30
GEM2108 Prozedurale Modellierung .....	32
GEM2109 Kryptographie .....	34
GEM5100.1 Projektarbeit .....	36
GEM5100.2 Wissenschaftliche Ausarbeitung zu Projekt .....	37
GEM6101 Masterarbeit .....	38

# 1 Ziele und Aufbau des Studiengangs Game Engineering und Visual Computing

Die Ziele des Studiengangs "Game Engineering und Visual Computing" sind die Vermittlung eines tiefen Verständnisses und der Analyse-, Design-, Adaption- und Umsetzungskompetenz von Verfahren und Algorithmen aus den Bereichen Game Engineering, Computer Vision, Echtzeitrendering und weiteren Bereichen des Visual Computing. Studierende erlangen die Fähigkeit zur Auswahl, Adaption, Kombination und Anwendung der oben genannten Verfahren für konkrete Anforderungen mit hoher Komplexität und praktischer Relevanz. Diese Kompetenzen werden in Master-Projekten, die die wissenschaftlich-abstrakte Analyse mit der Anwendung verbinden, ganz gezielt und spezifisch gefördert. Hierbei wird unabhängig von den fachlichen Kenntnissen und Fähigkeiten eine Methodenkompetenz entwickelt, die die Studierenden allgemein befähigt, forschungsnahe, wissenschaftliche Herangehensweisen auf konkrete Anwendungen zu übertragen.

Die Nachfrage nach immersiven, teil- oder vollsynthetischen Visualisierungen und nach Anreicherungen realer Videos mit synthetischen Teilen steigt rasant. Die genannten Themengebiete stellen heute neben der Unterhaltungsindustrie auch solche Unternehmen, die nicht in der Unterhaltungs- und Medienindustrie tätig sind, immer mehr vor die Herausforderung häufig aus Computerspielen bekannte, hochkomplexe, forschungsnahe Methoden in ihre Produkte oder Unternehmensprozesse zu integrieren. Die hierfür erforderlichen Kenntnisse und Fertigkeiten vermitteln die Pflichtmodule des Studiengangs. In deren praktischen Teilen, dem Master-Projekt sowie den Wahlpflichtmodulen erlangen die Studierenden darüber hinaus verschiedene, tiefergehende Kompetenzen.

Der Masterstudiengang schließt an einen vorausgegangenen (Bachelor-) Studiengang Informatik - Game Engineering oder einen Informatik-Studiengang mit entsprechender Schwerpunktbildung in den Fächern

- Programmieren in C++
- Computergrafik
- Game Engineering

an. Auf Basis dieser Grundausbildung erlangen die Studierenden Kompetenzen, über die in unten stehender Tabelle ein Überblick gegeben wird:

Nr.	Studienziele	Lernergebnisse
1	Algorithmische und mathematische Kompetenz	Verständnis aktueller mathematischer Konzepte und Algorithmen Fertigkeit zur Auswahl und Adaption von aktuellen Vorgehensweisen für gegebene Problemstellungen Kompetenz zur wissenschaftlich fundierten Weiterentwicklung
2	Methodenkompetenz Game Engineering und Visual Computing	Tiefes Verständnis der Algorithmen und Methoden in den Bereichen Computer Vision, voll- und teilsynthetische Visualisierung und Game Engineering Fertigkeit zur Auswahl und Adaption von aktuellen Vorgehensweisen für gegebene Problemstellungen Kompetenz zur Erarbeitung einer Lösung für gegebene, komplexe Problemstellungen aus oben genannten Bereichen Kompetenz zur wissenschaftlich fundierten Weiterentwicklung von Technologien dieser Bereiche

3	Technologiekompetenz	Kenntnis aktueller Hard- und Software im Bereich hochqualitativer 3D-Visualisierung und Computer Vision sowie deren algorithmischer Grundlagen Fähigkeit zur Auswahl und Kombination passender Lösungen Kompetenz zur Erarbeitung einer Gesamtlösung
4	Anwendungskompetenz	Fähigkeit das erworbene Wissen auch in großen und fachlich komplexen Problemstellungen einzusetzen und erfolgreich anzuwenden Befähigung zur wissenschaftlichen Arbeit Verständnis für den Zusammenhang der verschiedenen Fachbereiche Kompetenz zur Kombination und Verknüpfung von Problemlösungen verschiedener Fachbereiche des Studiengangs zur Lösung komplexer Problemstellungen
5	Soziale und überfachliche Kompetenzen	Kenntnis der verwandten Vorgehensweisen im Game Design Fähigkeit zur Kommunikation und Realisierung der Schnittstelle zwischen Game Engineering und Game Design Kompetenz größere Projekte des Fachgebiets zu planen, zu organisieren und zu leiten, sowie gegenüber Dritten zu vertreten Probleme frühzeitig zu erkennen und zu lösen und Projekte erfolgreich abzuschließen Fähigkeit, im Team fachlich als auch leitend verantwortliche Funktionen zu übernehmen Kompetenz gemischte Teams zu koordinieren und zu leiten, um ein Gesamtprojekt zu realisieren

Ein wesentlicher Teil des Kompetenzerwerbs und der Lernergebnisse wird durch die Pflichtmodule Algorithmen für Realtime Rendering, Computer Vision, Advanced Game Engineering, Simulation und Datenvisualisierung und Augmented Reality erreicht. Das Verständnis der Algorithmen der Computer Vision, insbesondere der projektiven Geometrie für Ein- und Zweikamerasysteme, bildet die Basis für geeignete Lösungsansätze zur Registrierung, zur Visualisierung und zur Interaktion in einem Anwendungskontext von Augmented Reality Szenarien. Um solche Szenarien in Echtzeit-Anwendungen auf aktueller Grafikhardware realisieren zu können, erlernen und verstehen die Studierenden im Modul Algorithmen für Realtime Rendering Vorgehensweisen und mathematische Hintergründe moderner Grafikanwendungen wie Interpolationsverfahren, Problemen der algorithmischen Geometrie und können diese situationsadäquat auswählen, adaptieren und anwenden. An der Schnittstelle zum Game Design können die Studierenden durch das Modul Advanced Game Engineering Animationen, Farb- und Formgebungen für Computerspiele richtig einschätzen und entwerfen Game-Levels vom psychologischen Standpunkt her optimal für eine dedizierte Zielgruppe. Im Modul Simulation und Datenvisualisierung arbeiten die Studierenden mit verschiedenen Arten von Daten aus unterschiedlichen Aufgabenfeldern, z.B. der Bildgebung in der Medizin. Sie sammeln Erfahrung mit ausgewählten Algorithmen der künstlichen Intelligenz zur Visualisierung und Simulation, wie z.B. durch Gruppen autonom gesteuerter Agenten.

Der Zusammenhang zwischen den übergeordneten Zielen und den Lernergebnissen des Masterstudiengangs sowie der Beitrag der Pflicht- und Wahlpflichtmodule zur Umsetzung dieser Ziele ist in der folgenden Zielmatrix dargestellt:

Modul	Studienziel				
	1	2	3	4	5
Computer Vision	++	++	+	+	
Augmented Reality	++	++	++	+	
Advanced Game Engineering	+	+	+	++	++
Algorithmen für Realtime Rendering	++	++	+	+	
Simulation und Datenvisualisierung	++	++		++	+
Wissenschaftliche Ausarbeitung zu Projekt	++	++		+	++
Projekt	+	+	+	++	++
Parallelprogrammierung	++	+	++	+	
Algorithmen und Strategien zur Entscheidungsunterstützung	++			++	+
Deep Learning	+	++	++	++	
Serious Games	+		+	++	++
Masterarbeit	+	++	+	++	+
Physically Based Rendering	++	++		+	
Mobile Roboter	++		+	++	

Das Masterstudium ist als Voll- oder Teilzeitstudium konzipiert. Das Vollzeitstudium umfasst einschließlich der Masterarbeit drei Semester. Die beiden ersten Semester beinhalten die theoretische Ausbildung. Bei den Modulen wird zwischen Pflicht- und Wahlpflichtmodulen unterschieden. Zu den Pflichtmodulen zählt auch ein Projekt, das aus einem praktischen Teil und einem wissenschaftlich-theoretischen Teil besteht. Ziel dieser Kombination ist die gezielte Förderung der Kompetenz, forschungsnahe Themen zu analysieren, zu adaptieren und weiterzuentwickeln. Die jeweilige Aufgabe wird im praktischen Teil des Projekts jeweils auch in einem konkreten Anwendungsszenario umgesetzt. Der wissenschaftliche Teil des Projekts hat dabei in etwa den Charakter eines wissenschaftlichen Seminars, wobei das Thema auf Grund der praktischen Anforderungen selbst erschlossen werden muss. Dieser Baustein ist charakteristisch und besonders wichtig für diesen Studiengang, da er dessen Fokus auf angewandte Wissenschaften in diesem komplexen Umfeld weiter betont. Die zeitliche Lage des Projektes kann bei Studienbeginn zum Sommersemester im ersten Studiensemester liegen. Dies ist unproblematisch, da durch den wissenschaftlich-theoretischen Teil die Grundlagen für das Projekt auf Basis der geforderten Voraussetzungen an die Studierenden gezielt erarbeitet werden.

Das dritte Semester dient der Anfertigung einer Masterarbeit, die im Interesse einer raschen Praxiseingliederung der Studierenden entweder im Rahmen eines Projektes mit einem Partner aus der Industrie oder in einem der Forschungsprojekte der Hochschule erstellt wird. In ihr sollen die Studierenden ihre Fähigkeit nachweisen, die im Studium erworbenen Kenntnisse, Fertigkeiten und Kompetenzen in einer selbstständig angefertigten, anwendungsorientiert-wissenschaftlichen Arbeit auf komplexe Aufgabenstellungen anzuwenden.

## Aufbau des Studiengangs Game Engineering und Visual Computing

Der Masterstudiengang ist modularisiert. In Anlehnung an das European Credit Transfer System (ECTS) werden für die drei Semester des Masterstudiums insgesamt 90 Leistungspunkte (CP) vergeben, und zwar pro Semester 30 Leistungspunkte. Insgesamt sind 25 CP für Pflichtmodule vorgesehen, 20 Leistungspunkte für Wahlpflichtmodule, 15 Leistungspunkte für Projekte mit wissenschaftlichen Arbeiten und 30 Leistungspunkte für die Masterarbeit. Das Studium

kann sowohl im Winter- als auch im Sommersemester begonnen werden. Wird das Studium im Wintersemester begonnen, ändert sich nur die Abfolge der Semester. Durch die Konzeption der Module als abgeschlossene Einheiten entsteht daraus für die Studierenden kein Nachteil.

### Game Engineering und Visual Computing, Master (M. Sc.) - Vollzeit

Hochschule für angewandte Wissenschaften Kempten

gültig mit Studienbeginn WS 2017/18

Semester

3	Masterarbeit																														ECTS-Punkte
2	Computer Vision	Advanced Game Engineering	Augmented Reality	WP-Modul	WP-Modul	WP-Modul																									
1	Algorithmen für Real Time Rendering	Simulation und Datenvisualisierung	WP-Modul	Wissenschaftliche Ausarbeitung zu Projekt	Master-Projekt																										
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	

Legende:

- Pflichtmodule
- Wahlpflichtmodule
- Projekt (Modul jeweils aus zwei Teilmodulen)
- Masterarbeit

Das Masterstudium kann auch als Teilzeitstudium in sechs Semestern absolviert werden. Im Teilzeitstudium sind pro Semester 15 Leistungspunkte zu erbringen. In den ersten vier Semestern erfolgt die theoretische Ausbildung und die beiden letzten Semester sind für die Masterarbeit reserviert.

### Game Engineering und Visual Computing □ Master (M. Sc.) - Teilzeit

Hochschule für angewandte Wissenschaften Kempten

gültig mit Studienbeginn WS 2017/2018

Semester

6	Masterarbeit															ECTS-Punkte
5	Masterarbeit															
4	WP-Modul	WP-Modul	WP-Modul													
3	Wissenschaftliche Ausarbeitung zu Projekt	Master-Projekt														
2	Computer Vision	Advanced Game Engineering	Augmented Reality													
1	Algorithmen für Real Time Rendering	Simulation und Datenvisualisierung	WP-Modul													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

Legende:

- Pflichtmodule
- Wahlpflichtmodule
- Projekt (Modul jeweils aus zwei Teilmodulen)
- Masterarbeit

Für den Studiengang Game Engineering und Visual Computing stehen folgende Ansprechpartner zur Verfügung:

Studiengangkoordinator: Prof. Dr. Bernd Dreier  
 Studienfachberater: Prof. Dr. Bernd Dreier  
 Vorsitzender der Prüfungskommission: Prof. Dr. Ulrich Göhner

## 2 Begriffserläuterungen

### ECTS - European Credit Transfer System

Diese Vereinbarungen zur Anrechnung, Übertragung und Akkumulierung von Studienleistungen basieren auf dem Arbeitspensum, das Studierende durchzuführen haben, um die Ziele des Lernprogramms zu erreichen. Für jede studienbezogene Leistung wird der voraussichtliche durchschnittliche Arbeitsaufwand angesetzt und auf das Studienvolumen angerechnet. Der Arbeitsaufwand umfasst Präsenzzeit und Selbststudium ebenso wie die Zeit für die Prüfungsleistungen, die notwendig sind, um die Ziele des vorher definierten Lernprogramms zu erreichen. Mit dem ECTS können Studienleistungen international angerechnet und übertragen werden.

### Arbeitsaufwand (Workload) und Leistungspunkte (ECTS-LP)

Der Arbeitsaufwand der Studierenden wird im ECTS in credit points angegeben. Deutsche Übersetzungen für credit point sind die Begriffe Leistungspunkt oder ECTS-Punkt. Ein Arbeitsaufwand von 30 Zeitstunden bedeutet einen Leistungspunkt. Der Arbeitsaufwand von Vollzeitstudierenden entspricht 60 Leistungspunkten pro Studienjahr, also 30 Leistungspunkten pro Semester. Das sind 1.800 Stunden pro Jahr oder 45 Wochen/Jahr mit 40 Stunden/Woche.

Der Arbeitsaufwand setzt sich zusammen aus:

- Präsenzzeit
- Zeit für die Vor- und Nachbereitung des Vorlesungsstoffs,
- Zeit für die Vorbereitung von Vorträgen und Präsentationen,
- Zeit für die Erstellung eines Projekts,
- Zeit für die Ausarbeitung einer Studienarbeit,
- Zeit für notwendiges Selbststudium,
- Zeit für die Vorbereitung auf mündliche oder schriftliche Prüfungen.

Die Bachelorstudiengänge mit sieben Semestern bescheinigen erfolgreichen Studierenden 210 ECTS-LP, die dreisemestrigen Masterstudiengänge weitere 90 ECTS-LP. Damit ist die Forderung nach 300 ECTS-LP für ein erfolgreich abgeschlossenes Masterstudium erfüllt.

### Semesterwochenstunden und Präsenzzeit

Eine Semesterwochenstunde ist die periodisch wiederkehrende Lehreinheit in einem Modul, in der Regel im Rhythmus von einer oder zwei Wochen. Dabei wird eine Präsenz von 45 Minuten plus Wegzeiten gerechnet, sodass die Vorlesungsstunde als eine Zeitstunde gewertet wird. Wir rechnen mit einer Vorlesungszeit von 15 Wochen pro Semester, wodurch sich aus der Zahl der Semesterwochenstunden die geforderte Präsenzzeit ("Kontaktzeit") direkt ableitet: 1 SWS entspricht 15 Stunden Präsenzzeit.

### Module

Der Studiengang setzt sich aus Modulen zusammen. Ein Modul repräsentiert eine inhaltlich und zeitlich zusammengehörige Lehr- und Lerneinheit. Module werden in der Regel in einem

Semester abgeschlossen. Modulgruppen sind Zusammenfassungen von Modulen mit einem weiteren inhaltlichen Zusammenhang. In allen Fällen stellt ein Modul oder ein Teilmodul eine Einheit dar, für die innerhalb und am Ende eines Semesters eine Prüfungsleistung erbracht werden kann, für die Leistungspunkte vergeben werden. Die Lehrveranstaltungen werden derzeit in deutscher Sprache gehalten.

## **Studienbegleitende Prüfungen und Studienfortschritt**

Sämtliche Prüfungen erfolgen über das gesamte Studium verteilt studienbegleitend und stehen in direktem Bezug zur Lehrveranstaltung. Prüfungsbestandteile können je nach Lehrveranstaltung begleitend oder nach Abschluss des Moduls stattfinden, beispielsweise als Referat, Klausurarbeit, mündliche Prüfung, Hausarbeit mit Kolloquium, Entwurf mit Kolloquium, Laborbericht, Exkursionsbericht oder einer Kombination. In den Beschreibungen der einzelnen Module wird im Modulhandbuch die jeweilige Prüfungsform festgelegt. Eine Wiederholung der Prüfung eines Moduls erfolgt bei Nichtbestehen im folgenden Semester.



## 3 Modulbeschreibungen

### GEM1101 Algorithmen für Real Time Rendering

#### Allgemeines

<b>Modulverantwortliche(r):</b>	Prof. Dr. Bernd Dreier
<b>Dozent(en):</b>	Prof. Dr. Bernd Dreier
<b>Modultyp:</b>	Pflichtmodul
<b>Voraussetzungen:</b>	Grundlegende Kenntnisse der OpenGL-Rendering Pipeline entsprechend dem im Bachelor Informatik - Game Engineering angebotenen Fach
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Sommersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS betreutes Praktikum in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Leistungsnachweise im Praktikum, Leistungsnachweise sind Zulassungsvoraussetzung schriftl. Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

#### Lernergebnisse und Inhalte

##### Lernergebnisse:

- Die Studierenden haben ein vertieftes Verständnis der Rendering-Pipeline
- Die Studierenden verstehen die Anforderungen neuerer OpenGL-Versionen (3.x, 4.x) und können dafür komplexe Shader auf der Basis von GLSL entwickeln
- Die Studierenden können komplexe Echtzeit-Anwendungen mit OpenGL realisieren
- Die Studierenden kennen Vor- und Nachteile verschiedener Arten der Speicherung polygonaler Netze und können diese auswählen und implementieren
- Die Studierenden verstehen die mathematischen Hintergründe moderner Grafikanwendungen
- Die Studierenden verstehen Interpolationsverfahren wie z.B. NURBS und können diese in der Computergrafik und Games anwenden
- Die Studierenden kennen ausgewählte Algorithmen der algorithmischen Geometrie und deren Anwendung für Computergrafik und Games

##### Lehrinhalte:

- Shaderprogrammierung mit GLSL (Stand OpenGL 4.1)
- OpenGL Puffer und Off-Screen-Rendering

- Bump-/Shadowmapping, TBN-Koordinatensystem
- Perspektivisch korrekte Interpolation
- Speicherung und algorithmische Behandlung polygonaler Netze
- Bézier-Kurven, Splines und NURBS
- Neue Konzepte in OpenGL 3/4
- Bounding Volumes und ausgewählte Algorithmen der algorithmischen Geometrie

**Literatur:**

- Eric Lengyel, Mathematics for 3D Game Programming and Computer Graphics, 3rd Edition, 2011
- Edward Angel, Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, Addison-Wesley, 2011
- Akenine-Möller, Real-Time Rendering, AK Peters, 2008
- Computational Geometry, Mark de Berg, et al., Springer, 2008

## GEM1102 Simulation und Datenvisualisierung

### Allgemeines

Die Lehrveranstaltung vermittelt Techniken zur Implementierung der Simulation von Prozessen und physikalischen Phänomenen sowie Methoden der wissenschaftlichen Visualisierung zur intuitiven Darstellung und visuellen Analyse von großen Datenmengen.

<b>Modulverantwortliche(r):</b>	Prof. Dr. Christoph Bichlmeier
<b>Dozent(en):</b>	Prof. Dr. Christoph Bichlmeier
<b>Modultyp:</b>	Pflichtmodul
<b>Voraussetzungen:</b>	Grundlegende Kenntnisse zu der OpenGL-Rendering Pipeline und der Computergraphik sowie in der C++ Programmierung
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Sommersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übungen (verpflichtend)
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übungen/Praktikum 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Erfolgreiche Teilnahme an den Übungen, schriftl. Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	keine Hilfsmittel

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Nach erfolgreicher Teilnahme an der Lehrveranstaltung kennen die Studierenden verschiedene Anwendungsbereiche der Simulation und Visualisierung. Sie können für einige diese Anwendungsbereiche selbständig die erlernten Algorithmen implementieren.

#### Lehrinhalte:

Die Studierenden können mithilfe von Excel, C++, Qt, OpenGL und GLSL Lösungen für folgende Themenbereiche realisieren.

- Wissenschaftliche Durchführung einer Datenerhebung und Datenauswertung
- Visuelle Darstellung von statistischen Datenerhebungen
- Visualisierung von Volumendaten und medizinischen Schnittbildern
- Visualisierung von Strömungsdaten
- Simulation von Weichgewebe
- Implementierung und Visualisierung von KI-Algorithmen zur Simulation zur Nachbildung von menschlichem Verhalten

#### Literatur:

- Artificial Intelligence for Games; 18. September 2009 von Ian Millington , John Fun-

---

ge

- Visual Computing for Medicine (Second Edition)□Theory, Algorithms, and Applications; Bernhard Preim and Charl P. Botha
- Visualisierung: Grundlagen und allgemeine Methoden; 2013; Heidrun Schumann, Wolfgang Müller
- Diverse Paper aus verschiedenen Quellen (Fachzeitschriften und Konferenzbände)

## GEM1103 Computer Vision

### Allgemeines

<b>Modulverantwortliche(r):</b>	Prof. Dr. Klaus Ulhaas
<b>Dozent(en):</b>	Prof. Dr. Klaus Ulhaas
<b>Modultyp:</b>	Pflichtmodul
<b>Voraussetzungen:</b>	Englischkenntnisse
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Wintersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Schriftliche Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	nicht programmierbarer Taschenrechner

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Die Studierenden sind in der Lage

- den theoretischen Hintergrund ausgewählter Verfahren der Computer Vision und der OpenCV zu erklären.
- die gelernten Verfahren zu kombinieren und mit der OpenCV in C++ auf Beispiele aus der Praxis zu übertragen.
- die projektive Geometrie für Ein- und Zweikamerasysteme zu verstehen, sowie die grundlegenden Schritte zur Triangulation wiederzugeben und zu berechnen

#### Lehrinhalte:

Folgende Inhalte werden in der Veranstaltung bearbeitet und an praktischen Beispielen implementiert:

- Einführung in OpenCV
- 2D Computer Vision:
- Segmentierung,
- Filteroperationen,
- Objekterkennungsmethoden
- Merkmalerkennung
- Homographie-Abbildungen
- 3D Computer Vision:
- Kamerakalibration,

- 3D Szenengeometrie

**Literatur:**

Literatur und Empfehlungen zu Einzelthemen werden fortlaufend in der Vorlesung bekanntgegeben

- age, 2014.

## GEM1104 Advanced Game Engineering

### Allgemeines

<b>Modulverantwortliche(r):</b>	Prof. Dr. Tobias Breiner
<b>Dozent(en):</b>	Prof. Dr. Tobias Breiner
<b>Modultyp:</b>	Pflichtmodul
<b>Voraussetzungen:</b>	
<b>Verwendbarkeit:</b>	Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Wintersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übungen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übungen 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Schriftl. Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	keine Hilfsmittel

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage,

- Animationen, Farb- und Formgebungen für Computerspiele richtig einzuschätzen und ein Game-Level vom psychologischen Standpunkt her optimal für eine dedizierte Zielgruppe zu entwerfen.
- aufbauend auf den Level Design-Vorlagen eine 3D-Szene eigenständig und zu programmieren.
- Szenen- bzw. Levelobjekte hierarchisch optimal zu organisieren.

#### Lehrinhalte:

- Physiologie des visuellen Systems, inklusive biologischer Grundlagen
- Psychologie der Bewegung und deren Einsatz in Computerspielen
- Gestaltgesetze
- Psychologie der Form und deren Einsatz in Computerspielen
- Psychologie der Farbe und deren Einsatz in Computerspielen
- Physiognomie und Anatomie für die Erstellung von Spiele-Charakteren
- Level-Design
- Optimaler Aufbau von Szenenhierarchien
- Lade- und Speicherhierarchien für 3D-Szenen
- Sound, Licht und Schatten in 3D-Games
- Texturerstellung, Texturierung, UV-Mapping, etc. in der Praxis (optional)

- 
- Hardware-Konfigurationen (optional)
  - Programmierung von Echtzeitanwendungen
  - Verteiltes Rendering, Stereoskopie, Multiskopie (optional)
  - Partikelsysteme (optional)
  - Split-Screens und In-Screens
  - Viewport-Stile
  - Spielelogik und Animationen

**Literatur:**

- Vektoria-Manual (downloadbar unter [www.vektoria-engine.com](http://www.vektoria-engine.com))
- Breiner, Tobias: Game-Psychologie, Springer-Verlag, 2018 (erhältlich ab Sommer 2018)



## GEM1105 Augmented Reality

### Allgemeines

<b>Modulverantwortliche(r):</b>	Prof. Dr. Klaus Ulhaas
<b>Dozent(en):</b>	Prof. Dr. Klaus Ulhaas / Prof. Dr.-Ing. René Bühling
<b>Modultyp:</b>	Pflichtmodul
<b>Voraussetzungen:</b>	Englischkenntnisse
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Wintersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Praktikum 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	erfolgreiche Teilnahme an den Übungen schriftl. Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	nicht programmierbarer Taschenrechner

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Die Studierenden kennen die Anforderungen von Augmented Reality Anwendungen im Indoor- und Outdoor-Bereich.  
Bestimmte Lösungsansätze zur Registrierung der virtuellen mit der physikalischen Welt sind Ihnen bekannt.  
Die Studierenden sind in der Lage

- den Begriff Augmented Reality zu erklären und Anwendungen im Bereich der Mixed Reality einzuordnen.
- die Bestandteile und die Anforderungen und sowie die technischen Möglichkeiten eines Augmented Reality Systems zu analysieren.
- eine einfache Augmented Reality Anwendung mit Video-See-Through-Techniken zu implementieren.
- geeignete Lösungsansätze zur Registrierung, zur Visualisierung und zur Interaktion in einem Anwendungskontext auszuwählen und diese zu implementieren.
- eine Augmented Reality Anwendung mithilfe aktueller Hardware zu entwickeln
- Anforderungen und Aufgabenstellungen in ausgewählten Anwendungsbereichen zu benennen

#### Lehrinhalte:

Durch Methoden der Augmented Reality (AR) wird die menschliche Wahrnehmung erweitert, indem virtuelle Informationen nahtlos in reale Welten integriert werden. Das Modul beinhaltet Softwaretechniken aus den Bereichen der Computergrafik und der Bild- und Signalverarbeitung, sowie Hintergründe zur Multimediakonzeption. Es sind folgende The-

men vorgesehen:

- Überblick über Techniken und Definition von Augmented Reality
- Numerische Optimierungsverfahren für den Einsatz in Echtzeitsystemen
- User Motion Tracking und Registrierung von Objekten
- Relativ und absolut arbeitende Inside-Out bzw. Outside-In Systeme zur Registrierung
- Visuelle Ausgabe
- Interaktion
- AR-Anwendungen (Outdoor und Indoor)
- Wahrnehmungsaspekte und Informationsvisualisierung
- Gestaltung visueller Inhalte
- Interaktionsdesign

#### **Literatur:**

IEEE Proceedings zum International Symposium on Mixed-and Augmented Reality

- Gray Bradski & Adrian Kaehler, Learning OpenCV, O'REILLY, 2008
- Baggio, Escrivá, Mahmood, Shilkrot, Emami, Ievgen, Saragih, Mastering OpenCV with Practical Augmented Reality Projects, Packt Published Ltd., 2012
- Ralf Dörner et al.: Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität, Springer, 2014
- D. Schmalstieg, T. Höllerer, Augmented Reality: Principles and Practice, Addison Wesley, 2016, ISBN-13: 978-0321883575
- Alan B. Craig, Understanding Augmented Reality: Concepts and Applications, Morgan Kaufmann, 2013, ISBN-13: 978-0240824086
- Siltanen, Sanni. 2012. Theory and applications of marker-based augmented reality. Espoo, VTT. 199 p. + app. 43 p. VTT Science; ISBN 978-951-38-7449-0 (soft back ed.) 978-951-38-7450-6 (PDF), <http://www.vtt.fi/inf/pdf/science/2012/S3.pdf>

## GEM2101 Serious Games

### Allgemeines

Der Themenbereich Serious Games untersucht die Möglichkeit mithilfe von Computerspielen und Simulationen Fachwissen zu vermitteln, die Augen für kulturelle und gesellschaftliche Themen zu öffnen, berufliche Fertigkeiten zu schulen und Heilungsprozesse zu beeinflussen. Weitere Zielsetzungen sind möglich. Serious Games machen es sich zur Aufgabe neben der Unterhaltung des Spielers auch ernsthafte Botschaften mithilfe des Spiels/der Simulation zu vermitteln. Neue Spielkonzepte sollen didaktische Erkenntnisse und Technologie der Computerspielwelt verknüpfen. Weiterhin werden Serious Games genutzt, um große Datenmengen über Spielerverhalten zu erheben, die für wissenschaftliche Zwecke von Nutzen sind. Neben dem Spieldesign befasst sich ein Teil der Forschung mit der Datenerhebung und der Auswertung dieser Daten um zu überprüfen, ob das „seriöse“ Ziel des jeweiligen Spiels bzw. der jeweiligen Simulation erfüllt wird. Darunter fällt beispielsweise die Vermessung des Lerneffekts, des Einflusses auf die Genesung oder der Änderung der persönlichen Einstellung zu einem gesellschaftlichen Thema.

<b>Modulverantwortliche(r):</b>	Prof. Dr. Christoph Bichlmeier
<b>Dozent(en):</b>	Prof. Dr. Christoph Bichlmeier
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	keine
<b>Verwendbarkeit:</b>	Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Sommersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übungen (verpflichtend)
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	erfolgreiche Teilnahme an den Übungen schriftl. Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	keine Hilfsmittel

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Nach Teilnahme an dem Fach Serious Games können Studierende

- didaktische Konzepte und Lerntheorien in Zusammenhang mit Serious Games benennen und beschreiben
- sich in ein interdisziplinäres Feld einarbeiten und einen möglichen Mehrwert über Serious Games in diesem Fachbereich evaluieren
- ein Geschäftsmodell für ein Serious Game bewerten
- eine Benutzerstudie planen und durchführen um mögliche Effekte des Serious Games zu bewerten

- die Ergebnisse der Benutzerstudie statistisch auswerten
- grundlegende Erkenntnisse der Lernpsychologie benennen und beschreiben
- ein Konzept für ein didaktisch wertvolles Serious Game erstellen
- gute wissenschaftliche Praxis für das Studiendesign anwenden

**Lehrinhalte:**

Folgende Lerninhalte werden vermittelt

- Anwendungsbereiche Serious Games
- Interdisziplinäres Arbeiten
- Didaktische Konzepte für Games
- Lernpsychologie
- Studiendesign und Statistik

**Literatur:**

- Psychologie: 16., aktualisierte Auflage (Pearson Studium - Psychologie) Gebundene Ausgabe - 1. Juni 2004 von Philip G. Zimbardo und Richard J. Gerrig
- Beiträge des International Journal of Serious Games
- Beiträge der International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)
- Beiträge der Serious Play Conference
- Deskriptive Statistik, Statistische Methoden für Psychologen Teil 1, Christof Nachtigall, Markus Wirtz
- Wahrscheinlichkeitsrechnung und Inferenzstatistik, Statistische Methoden für Psychologen Teil 2, Christof Nachtigall, Markus Wirtz

## GEM2102 Deep Learning

### Allgemeines

<b>Modulverantwortliche(r):</b>	Prof. Dr. Jürgen Brauer
<b>Dozent(en):</b>	Prof. Dr. Jürgen Brauer
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	Analysis, Lineare Algebra, Programmierkenntnisse
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.), Studiengang Angewandte Informatik (M.Sc.)
<b>Angebot und Dauer:</b>	Wintersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übungen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Schriftliche Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	keine Hilfsmittel

### Lernergebnisse und Inhalte

#### Lernergebnisse:

- Die Studierenden kennen mindestens drei der wichtigsten DL Architekturen im Detail
- Für jeden dieser drei Ansätze haben Sie fundiertes Wissen über den Aufbau und Funktionsweise gesammelt und können dieses Wissen bei Problemstellungen im Bereich Bildverarbeitung und Spracherkennung durch Anwendung eines geeigneten DL Verfahrens umsetzen
- Die Studierenden verstehen, welche Eigenschaften die DL Verfahren besonders auszeichnet und von klassischen Machine Learning Verfahren unterscheidet
- Sie kennen die historische Entwicklung des DL Gebietes und können abschätzen welche weiteren Entwicklungen zu erwarten sind

#### Lehrinhalte:

##### Einführung

- Motivation Deep Learning: kürzliche Erfolge
- Geschichte des Deep Learnings
- Gründe für den Boom
- Das biologische Vorbild
- Technische Neuronenmodelle
- Einfache technische Neuronale Netze
- Perzeptron / Perzeptron-Lernregel

- Multi-Layer-Perzeptron / Backpropagation-Algorithmus
- Deep Learning Architekturen
- CNN: Convolutional Neural Networks für Bildverarbeitungsaufgaben
- LSTM: Long Short Term Memory zur Automatischen Spracherkennung und Sprachübersetzung
- GAN: Generative Adversarial Networks
- Neural Style Transfer
- Deep Learning Tricks
- Ausblick: wohin entwickelt sich der Ansatz?

**Literatur:**

- Yann LeCun, Yoshua Bengio, Geoffrey Hinton. Deep Learning. Nature, 2015.
- Jürgen Schmidhuber. Deep Learning in Neural Networks: An Overview. Neural Networks Journal 61 (2015): 85-117
- Softwarebibliothek TensorFlow: <https://www.tensorflow.org/>

## GEM2103 Algorithmen und Strategien zur Entscheidungsunterstützung

### Allgemeines

<b>Modulverantwortliche(r):</b>	Prof. Dr. Jochen Staudacher
<b>Dozent(en):</b>	Prof. Dr. Jochen Staudacher
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	Keine
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Wintersemester, ein Semester
<b>Lehrformen:</b>	3 SWS Seminaristischer Unterricht 1 SWS Praktikum in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	45 Stunden Präsenzzeit Vorlesung 15 Stunden Präsenzzeit Praktikum 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Die erfolgreiche Teilnahme an den Übungen gilt als Zulassungsvoraussetzung für eine 20-minütige mündliche Prüfung am Ende des Semesters.
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	keine Hilfsmittel

### Lernergebnisse und Inhalte

#### Lernergebnisse:

- Die Studierenden können entscheidungs- und spieltheoretische Konzepte auf praktische Fragestellungen aus Informatik und Wirtschaftswissenschaften anwenden.
- Sie können beschreiben und analysieren, welchen Einfluss die Struktur eines sozialen oder ökonomischen Netzwerks auf strategisches Handeln besitzt.
- Sie können für ausgewählte Anwendungsfälle aus großen Datensätzen sinnvolle Information extrahieren, die zur Entscheidungsunterstützung eingesetzt werden kann.
- Sie können die vorgestellten Methoden und Algorithmen in R umsetzen.
- Sie können zu den vorgestellten Ansätzen und Algorithmen zur Entscheidungsunterstützung kritisch Stellung nehmen und deren Potenziale und Grenzen charakterisieren.

#### Lehrinhalte:

- Entscheidungskriterien, Entscheidungsmodelle, Entscheidungsalgorithmen
- Nutzenfunktionen, rationales Entscheiden bei Risiko, Bernoulli-Prinzip
- Interaktives Entscheiden, Algorithmische Spieltheorie und Anwendungen in Informatik und Ökonomie
- Anpassung und Erweiterung spieltheoretischer Ansätze zur Analyse ökonomischer und sozialer Netzwerke

- Ökonomien mit Netzwerkeffekten
- Statistische Entscheidungstheorie
- Datenbasiertes Entscheiden: Algorithmen und Anforderungen an die Datenqualität

**Literatur:**

Es gibt nicht das EINE Lehrbuch zu dieser Vorlesung; verschiedene Teile der Vorlesung werden durch die folgenden Lehrbücher abgedeckt:

- James N. Webb: Game Theory: Decisions, Interaction and Evolution, Springer, 2007
- Michael Maschler, Eilon Solan, Shmuel Zamir: Game Theory, Cambridge University Press, 2013
- Hans-Jürgen Zimmermann: Operations Research, Vieweg, 2. Auflage, 2008
- Matthew O. Jackson: Social and Economic Networks, Princeton University Press, 2008
- David Easley, Jon Kleinberg: Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010
- Pierre Lafaye de Micheaux, Remy Drouilhet, Benoit Liquet: The R Software, Springer, 2014
- Larry Wasserman: All of statistics, Springer, 2004



## GEM2104 Parallelprogrammierung

### Allgemeines

<b>Modulverantwortliche(r):</b>	Prof. Dr. Ulrich Göhner
<b>Dozent(en):</b>	Prof. Dr. Ulrich Göhner
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	Keine
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Wintersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Schriftliche Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

### Lernergebnisse und Inhalte

#### Lernergebnisse:

- Die Studierenden kennen aktuelle parallele Rechnerarchitekturen einordnen und deren Vor- und Nachteile
- Sie verstehen die wichtigsten parallele Programmierkonzepte
- Sie können die Leistung von Parallelrechnern bei der Lösung praktischer Probleme bewerten
- Sie kennen die Vorgehensweisen bei der effektiven Parallelisierung von Algorithmen
- Sie beherrschen den effektiven Einsatz moderner paralleler Programmierwerkzeuge

#### Lehrinhalte:

- Einführung
- Leistungsbewertung, Amdahl und Gustafson-Gesetz
- parallele Rechnerarchitekturen
- Programmierkonzepte für Parallelrechner
- OpenMP und MPI
- GPU-Programmierung mit CUDA und OpenCL
- PRAM (Parallel Random Access Machine)
- Parallele Algorithmen und Komplexität
- Präfixberechnungen
- Parallele Berechnung numerischer Ausdrücke

- Parallele Matrix- und Sortieralgorithmen
- Grid- und Cloud-Computing
- Quantenrechner

**Literatur:**

- Tanenbaum, A.S.: "Computerarchitektur. Strukturen - Konzepte - Grundlagen", Pearson Studium, 2006.
- "OpenMP Application Program Interface", Version 4.0 - RC 1, November 2012.
- "MPI: A Message-Passing Interface Standard", Version 3.0, Message Passing Interface Forum, September 21, 2012.
- "CUDA C PROGRAMMING GUIDE", NVIDIA, 2012.
- "Introduction to OpenCL Programming", AMD, 2010.
- Homeister, M.: "Quantum Computing verstehen", Springer Fachmedien Wiesbaden, 2015.
- Balaji, P.: "Compute Unified Device Architecture," Programming Models for Parallel Computing , MITP, 2015.
- Rauber, Th.; Runger, G.: "Parallele Programmierung", Springer Vieweg Berlin Heidelberg, 2012.
- Rahm, E.; Sattler, K.-U.: " Verteiltes und Paralleles Datenmanagement", Springer Vieweg Berlin Heidelberg, 2015.

## GEM2105 Physically Based Rendering

### Allgemeines

<b>Modulverantwortliche(r):</b>	Hr. André Kettner
<b>Dozent(en):</b>	Hr. André Kettner
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	
<b>Verwendbarkeit:</b>	Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Wintersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS betreutes Praktikum in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Leistungsnachweise im Praktikum, Leistungsnachweise sind Zulassungsvoraussetzung schriftl. Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

### Lernergebnisse und Inhalte

#### Lernergebnisse:

- Die Studierenden kennen moderne Verfahren zur Umsetzung einer physikalisch basierten Render-Engine und sind in der Lage diese zu implementieren.
- Die Studierenden sind in der Lage, Szenenbestandteile wie Materialien und Licht realistisch zu definieren und verstehen die zugrundeliegenden theoretischen Konzepte von PBR-Engines.
- Die Studierenden sind in der Lage, in der Praxis gängige Optimierungsverfahren beim Rendering kontextbezogen zu beurteilen und softwareseitig umzusetzen.
- Die Studierenden kennen Methoden der Skalierbarkeit und Parallelisierbarkeit von Renderprozessen.
- Die Studierenden sind in der Lage, numerische Approximationsverfahren beim Rendering umzusetzen.

#### Lehrinhalte:

In der Lehrveranstaltung "Physically Based Rendering" (PBR) werden sowohl die theoretischen Konzepte, die einer modernen Render-Engine zugrundeliegen, als auch die praktische Umsetzung in Form eigener Implementierung, vermittelt. Neben der grundlegenden technischen Formulierung des Lichttransportproblems globaler Beleuchtung und dessen numerischen Approximation mittels Monte Carlo werden zahlreiche Optimierungsverfahren - wie sie sich in nahezu allen rezenten kommerziellen Renderern finden - betrachtet und angewandt.

- Überblick über aktuelle Rendermethoden und deren Einsatzbereiche

- Technischer Aufbau eines modernen Frameworks für realistische Bildsynthese
- Radiometrische Definition von Materialeigenschaften und Licht
- Theoretische Betrachtungen zur Simulation globaler Beleuchtung
- Numerische Approximation der Rendergleichung mittels Monte Carlo Integration
- Implementierung eines einfachen physikalisch basierten Renderers auf Monte Carlo Basis
- Sampling-Optimierung mittels stratified sampling und Quasi-Monte Carlo Methoden
- Umsetzung von Importance Sampling zur Varianzreduktion
- Pfadformulierung des Lichttransportproblems und Implementierung von path tracing
- Überblick über die alternativen Integratoren bi-directional path tracing und Metropolis Light Transport
- Überblick über gängige biased Verfahren wie Irradiance Caching und Photon Mapping
- Photometrische Bildrekonstruktionsmethoden

**Literatur:**

- Philip Dutré et al.: Advanced Global Illumination, A K Peters Ltd, 2006
- Matt Pharr, Greg Humphreys: Physically Based Rendering, Morgan Kaufmann, Nov. 2016

## GEM2106 Softwarearchitektur

### Allgemeines

<b>Modulverantwortliche(r):</b>	Prof. Dr. Georg Hagel
<b>Dozent(en):</b>	Prof. Dr. Georg Hagel
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	keine
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Sommersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Leistungsnachweis in der Übung, Schriftliche Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	keine Hilfsmittel

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage,

- Zu erklären, welche Faktoren Einfluss auf die Architektur einer Software haben.
- Zu beschreiben, welche Architekturprinzipien beim Entwurf einer Softwarearchitektur beachtet werden müssen.
- Bekannte Architekturmuster bei der Erstellung der Softwarearchitektur anzuwenden.
- Aspekte von Softwaresystemen zu beschreiben.
- Lösungsansätze für Aspekte von Softwaresystemen anzugeben.
- Softwarearchitekturen mit aktuellen Beschreibungsmethoden zu dokumentieren.
- Continuous Integration, Continuous Deployment und DevOps zu beschreiben und anzuwenden.

#### Lehrinhalte:

- Aufgaben von Softwarearchitekten
- Entwurf von Softwarearchitekturen
- Architekturmuster
- Architektur Aspekte
- Dokumentation von Softwarearchitekturen
- Continuous Integration, Continuous Deployment und DevOps

#### Literatur:

- 
- Posch, Thorsten et al.: Basiswissen Softwarearchitektur, dpunkt.verlag, 3. Auflage (2012)
  - Starke, Gernot: Effektive Software-Architekturen, Carl Hanser Verlag, 8. Auflage (2017)
  - Vogel, Oliver et al.: Software-Architektur, Elsevier (2005)
  - Buschmann, Frank et al.: POSA Band 1 - Pattern-orientierte Softwarearchitektur - Ein Pattern-System
  - Bass, Len et al.: Software Architecture in Practice, Third Edition, SEI Series in Software Engineering, Addison-Wesley (2012)
  - Oestereich, Bernd: Die UML 2.5 Kurzreferenz für die Praxis, Oldenbourg Verlag, 6. Auflage (2014)
  - Bass, Len et al.: DevOps: A Software Architect's Perspective, Addison Wesley (2015)

## GEM2107 Mobile Roboter

### Allgemeines

Mobile Roboter sind immer noch nicht in unserem Alltag allgegenwärtig. Zwar gibt es einige wenige Beispiele wie Staubsaugroboter, Drohnen, Marserkundungsroboter, autonome Fahrzeuge oder Fußballspielende Roboter beim RoboCup, doch meist handelt es sich bisher um Nischenlösungen. Wieso ist es so schwer mobile Roboter zu bauen, die Arbeiten übernehmen können, die heute nur von Menschen ausgeführt werden können? In der Vorlesung soll ein breiter Überblick über die verschiedenen Aspekte des Themas mobile Roboter erarbeitet werden. Nachdem ein Verständnis für die vielseitigen Herausforderungen der mobilen Robotik geschaffen wird, werden wir auf geeignete Sensorik und Aktorik für Roboter eingehen. Die Organisation der Informationsverarbeitung ist in einem mobilen Roboter ein wichtiges zentrales Thema. Entsprechend werden hier verschiedene Ansätze für sog. Roboterkontrollarchitekturen vorgestellt. Ein ebenso zentrales Thema ist das Thema der Navigation: wie kann ein mobiler Roboter robust durch eine bekannte oder sogar unbekannte Umgebung navigieren? Schließlich behandeln wir das Thema des Lernens von geeigneten Aktionen für Roboter. Hier haben sich verschiedene Ansätze wie z.B. das Reinforcement-Learning und das Imitationslernen etabliert. Am Ende der Vorlesung werden wir auf Besonderheiten der kooperativen Robotik eingehen und es werden Ansätze vorgestellt, die eine Zusammenarbeit von mehreren Robotern ermöglichen sollen. In den Übungen sollen zentrale Algorithmen anhand von Robotersimulatoren vertieft werden.

<b>Modulverantwortliche(r):</b>	Prof. Dr. Jürgen Brauer
<b>Dozent(en):</b>	Prof. Dr. Jürgen Brauer
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	Analysis, Lineare Algebra, Programmierkenntnisse
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Sommersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Leistungsnachweise in den Übungen Leistungsnachweise sind Zulassungsvoraussetzungen zur schriftlichen Prüfung Schriftliche Prüfung 90 Minuten am Ende des Semesters
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	keine Hilfsmittel

### Lernergebnisse und Inhalte

#### Lernergebnisse:

- Die Studierenden können einschätzen, welche Fähigkeiten heute von mobilen Robotern erwartet werden können und welche nicht
- Sie wissen welche Sensorik für eine robuste Umfelderkennung notwendig ist und

kennen Ansätze zur Umfelderkennung auf Basis geeigneter Sensordaten

- Sie haben fundiertes Wissen über verschiedene Ansätze von Roboterkontrollarchitekturen und kennen deren Vor- und Nachteile
- Die Studierenden kennen unterschiedliche Navigationsverfahren und können diese selber implementieren und anwenden
- Sie haben einen breiten Überblick über Verfahren die das Erlernen von Aktionen erlauben, kennen den theoretischen Hintergrund der Verfahren und können die Verfahren selber umsetzen
- Den Studierenden sind mögliche Auswirkungen der Verfügbarkeit intelligenter mobiler Roboter auf die Gesellschaft bewusst und können kritisch beurteilt werden

**Lehrinhalte:**

- Einführung: Geschichte der mobilen Roboter
- Herausforderungen für mobile Roboter
- Roboterkontrollarchitekturen
- Navigation, Lokalisation und Erstellung von Karten (SLAM), Pfadplanung
- Lernen von Aktionen, Reinforcement-Learning, Imitationslernen
- Kognitive Architekturen für Roboter
- Ausblick: wohin entwickelt sich der Bereich mobile Roboter?
- Mögliche Auswirkungen der Verfügbarkeit intelligenter mobiler Roboter auf die Gesellschaft

**Literatur:**

- Joachim Hertzberg et al. Mobile Roboter: Eine Einführung aus Sicht der Informatik Springer Vieweg Verlag. 2012.
- Roland Siegwart et al. Introduction to Autonomous Mobile Robots MIT. 2011



## GEM2108 Prozedurale Modellierung

### Allgemeines

Die prozedurale Erstellung von Game Levels und 3D-Simulationsumgebungen (z.B. zum Training von Neuronalen Netzen) wird in der nahen Zukunft an Bedeutung zunehmen. Dies liegt darin begründet, dass die Komplexität der 3D-Szenen kontinuierlich in Zusammenhang mit den Ansprüchen der User und der Leistungsfähigkeit der Grafikkarten zunimmt, so dass die traditionelle manuelle Erstellung von Geometrien mittels Modellierungssoftware (Blender, Maya, 3D Studio Max, Rhino, Softimage, etc.) damit nicht mehr Schritt halten kann. Prozedurale Modellierung kann dagegen beliebig polygonal skaliert und parametrisiert werden.

In diesem Modul lernen die Studierenden daher verschiedene Algorithmen und Verfahren zur prozeduralen Geometrieerzeugung kennen. Sie trainieren darüber hinaus, diese theoretischen Methoden in der Praxis anzuwenden und die damit erzeugten Geometrien in sinnfälliger Weise zu einem Game-Level zu arrangieren.

<b>Modulverantwortliche(r):</b>	Prof. Dr. Tobias Breiner
<b>Dozent(en):</b>	Prof. Dr. Tobias Breiner
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	keine
<b>Verwendbarkeit:</b>	Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Sommersemester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Unterricht, 30 Stunden Präsenzzeit Übung, 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	90-minütige schriftliche Klausur.  Alternativ ist im Falle von Lockdowns bei Pandemien oder vergleichbaren Hindernissen auch eine elektronische Abgabe eines prozedural erstellten Game-Levels zur Benotung möglich.
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Die Studierenden sind in der Lage:

- div. virtuelle Objekte (Pflanzen, Gebäude, Straßnetze, Terrain, Steine, Planeten, technische Geräte etc.) prozedural und parametrisiert zu erstellen.
- ein vollständiges Game-Level selbstständig zu erstellen, und dabei vorwiegend prozedurale Methoden zu verwenden.
- die Mathematik hinter der prozeduralen Modellierung zu verstehen und sie sinnvoll anzuwenden.

---

## **Lehrinhalte:**

Inhalte sind:

- Vor- und Nachteile prozeduraler Erstellung, Beispiele in Computerspielen
- Arten von polygonalen Geometrien, Vor- und Nachteile derselben
- Modellieren mit Bézier-Patches
- Quadriken und parametrisierte Primitive
- Erweiterte Barr-Modellierung
- L-Systeme und ihre Anwendung zur Erzeugung verschiedener Pflanzenarten
- Hierarchitekturmodellierung
- Perlin Noise
- Terrain-Erstellung
- Planetenerstellung
- Optimaler Einsatz von prozeduralen Erzeugungsmethoden in Game-Levels
- Sweeps und Swakes
- komplexe Zahlen, Quaternionen, Biquaternionen, lokalsensitive homogene Matrizen und ihre Anwendungen in der Computergrafik

## **Literatur:**

- Vektoria-Manual, Downloadbar unter [www.games.hs-kempten.de](http://www.games.hs-kempten.de)

## GEM2109 Kryptographie

### Allgemeines

Wir beschäftigen uns mit verschiedenen gängigen Kryptographieverfahren (siehe Themenliste) sowie grundsätzlichen Fragestellungen rund um die Kryptographie: Woher kommt die Kryptographie? Was ist nötig um Kryptographie zu betreiben? Wann ist sie sicher? Wie wirkt sich Kryptographie (bzw. ihr Abwesenheit) auf die moderne Gesellschaft aus? Wie können Wahlen sicher durchgeführt werden? Wie können Kryptosystem analysiert/gebrochen werden? Welche Entwicklungen, die die Kryptographie wesentlich verändern werden, sind mittelfristig zu erwarten (Stichwort Quantencomputer)? Was gibt es für Ansätze um dem entgegenzuwirken?

<b>Modulverantwortliche(r):</b>	Prof. Dr. Elmar Böhler
<b>Dozent(en):</b>	Prof. Dr. Elmar Böhler
<b>Modultyp:</b>	Wahlpflichtmodul
<b>Voraussetzungen:</b>	Programmieren I+II, Algorithmen und Datenstrukturen, Theoretische Informatik, Diskrete Mathematik, Wahrscheinlichkeitsrechnung
<b>Verwendbarkeit:</b>	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	Wintersemester, ein Semester
<b>Lehrformen:</b>	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	30 Stunden Präsenzzeit Vorlesung 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
<b>Leistungsnachweis und Prüfung:</b>	Teilnahmepflicht in den Übungen, Schriftliche Prüfung 90 Minuten am Ende des Semesters. Erfolgreiche Teilnahme an den Übungen ist Zulassungsvoraussetzung.
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	handschriftliche Notizen, 1 DIN A4 Blatt, einseitig beschrieben, keine Kopie

### Lernergebnisse und Inhalte

#### Lernergebnisse:

#### Lehrinhalte:

Es werden die Grundlagen der Kryptographie behandelt, sowie einige der wesentlichsten verwendeten Verfahren. Mögliche Themen sind die folgenden:

- Historische Kryptosysteme und ihre Analyse
- Blockchiffren
- Betriebsarten
- DES
- AES

- Schlüsselaustausch
- Public-Key Kryptographie
- RSA
- Digitale Signaturen
- Primzahltests
- ElGamal
- Einführung in ECC
- Teilen von Geheimnissen
- Einblicke in die Quantenkryptographie
- Postquantenkryptographie
- Hashfunktionen
- Wahlverfahren
- Blockchains
- Pseudozufallszahlen
- Kryptographie in der Gesellschaft
- Zero-Knowledge Systeme
- Homomorphe Verschlüsselung
- Komplexitätstheoretische Grundlagen
- Berechnungsverifikation

**Literatur:**

- J. Buchmann: Einführung in die Kryptographie
- J. Rothe: Complexity Theory and Cryptology
- K. Wagner: Theoretische Informatik
- A. Menezes, P. van Oorschot, S. Vanstone: Handbook of Applied Cryptography

## GEM5100.1 Projektarbeit

### Allgemeines

<b>Modulverantwortliche(r):</b>	Professoren der Fakultät
<b>Dozent(en):</b>	Professoren der Fakultät
<b>Modultyp:</b>	Pflichtmodul
<b>Voraussetzungen:</b>	keine
<b>Verwendbarkeit:</b>	Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	jedes Semester
<b>Lehrformen:</b>	Projekt
<b>Leistungspunkte:</b>	10
<b>Arbeitsaufwand:</b>	15 Stunden Präsenzzeit Unterricht 285 Stunden selbständiges Arbeiten
<b>Leistungsnachweis und Prüfung:</b>	Projektbericht Präsentation

**Zur Prüfung zugelassene Hilfsmittel:**

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage,

- ein Projekt zu planen, zu organisieren, zu leiten und gegenüber Dritten zu vertreten
- Probleme im Projekt frühzeitig zu erkennen und zu lösen
- ein Projekt erfolgreich abzuschließen
- das erworbene Wissen auch in großen, fachlich komplexen, forschungsnahen Anwendungsszenarien einzusetzen und erfolgreich anzuwenden
- die Schnittstelle zwischen Game Engineering und Game Design zu realisieren und zu vertreten
- sich schnell in neue Anwendungsgebiete, Technologien und Grundlagen einzuarbeiten

#### Lehrinhalte:

projektabhängig

#### Literatur:

projektabhängig

## GEM5100.2 Wissenschaftliche Ausarbeitung zu Projekt

### Allgemeines

<b>Modulverantwortliche(r):</b>	Professoren der Fakultät
<b>Dozent(en):</b>	Professoren der Fakultät
<b>Modultyp:</b>	Pflichtmodul
<b>Voraussetzungen:</b>	keine
<b>Verwendbarkeit:</b>	Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	jedes Semester
<b>Lehrformen:</b>	
<b>Leistungspunkte:</b>	5
<b>Arbeitsaufwand:</b>	15 Stunden Präsenzzeit Vorträge und anschließender Diskussion 135 Stunden selbständiges Arbeiten, Vorbereitung der Präsentation, Ausarbeitung der Studienarbeit
<b>Leistungsnachweis und Prüfung:</b>	Studienarbeit und Kolloquium
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Nach dem erfolgreichen Besuch der Lehrveranstaltung sind die Studierenden in der Lage,

- sich in ein anspruchsvolles, wissenschaftliches Thema einzuarbeiten
- entsprechende Literatur zu recherchieren und aufzuarbeiten
- einen Vortrag zu einem wissenschaftlichen Thema zu erarbeiten, zu präsentieren und bei der nachfolgenden Diskussion zu vertreten
- eine schriftliche Ausarbeitung zu einem wissenschaftlichen Thema zu erstellen
- forschungsnahe Themen auf eine praktische Problemstellung anzuwenden und Lösungsvorschläge zu erarbeiten

#### Lehrinhalte:

themenabhängig

#### Literatur:

themenabhängig

## GEM6101 Masterarbeit

### Allgemeines

<b>Modulverantwortliche(r):</b>	Betreuender Professor
<b>Dozent(en):</b>	Betreuender Professor
<b>Modultyp:</b>	Pflichtmodul
<b>Voraussetzungen:</b>	Siehe Studien- und Prüfungsordnung
<b>Verwendbarkeit:</b>	Studiengang Game Engineering und Visual Computing (M.Sc.)
<b>Angebot und Dauer:</b>	jedes Semester
<b>Lehrformen:</b>	
<b>Leistungspunkte:</b>	30
<b>Arbeitsaufwand:</b>	900 Stunden
<b>Leistungsnachweis und Prüfung:</b>	Abschlussarbeit
<b>Zur Prüfung zugelassene Hilfsmittel:</b>	

### Lernergebnisse und Inhalte

#### Lernergebnisse:

Mit der Masterarbeit beweisen die Studierenden, dass sie komplexe, forschungsnahe Problemstellungen unter Einsatz der im Studium erworbenen Kenntnisse, Fertigkeiten und Kompetenzen mit wissenschaftlichen Methoden zu einer praktisch verwendbaren, konsistenten und überzeugenden Lösung führen können. Die Abschlussarbeit darf mit Zustimmung der Prüfungskommission in einer Einrichtung außerhalb der Hochschule ausgeführt werden.

#### Lehrinhalte:

#### Literatur: