



Fakultät Informatik

Studiengang Game Engineering und Visual Computing (Master)

Modulhandbuch

Stand: WS 2024/25

Juli 2024

Prof. Dr. E. Müller
Studiendekan der Fakultät Informatik

Prof. Dr. B. Dreier
Studiengangkoordinator

Prof. Dr. U. Göhner
Vorsitzender der Prüfungskommission

Inhaltsverzeichnis

1 Ziele und Aufbau des Studiengangs Game Engineering und Visual Computing	2
2 Begriffserläuterungen	6
3 Modulbeschreibungen	8
GEM1101 Algorithmen für Real Time Rendering	8
GEM1102 Datenvisualisierung	10
GEM1103 Computer Vision.....	12
GEM1104 Advanced Game Engineering	14
GEM1105 Augmented Reality	15
GEM2102 Deep Learning.....	17
GEM2103 Algorithmen und Strategien zur Entscheidungsunterstützung.....	19
GEM2105 Physically Based Rendering	21
GEM2106 Softwarearchitektur	23
GEM2107 Mobile Roboter.....	25
GEM2108 Prozedurale Modellierung	27
GEM2109 Kryptographie	29
GEM2110 3D Computer Vision	31
GEM2111 Big Data	33
GEM2112 Deep Learning für Computer Vision	35
GEM2114 Künstliche Intelligenz für Games - fortgeschrittene Themen	37
GEM5100.1 Projektarbeit	39
GEM5100.2 Wissenschaftliche Ausarbeitung zu Projekt.....	40
GEM6101 Masterarbeit.....	41

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-, Adaptions- 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 Grafikkhardware 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 Zielematrix dargestellt:

Modul	Studienziel				
	1	2	3	4	5
3D Computer Vision	++	+	+	++	+
Advanced Game Engineering	+	+	+	++	++
Algorithmen für Realtime Rendering	++	++	+	+	
Algorithmen und Strategien zur Entscheidungsunterstützung	++			++	+
Augmented Reality	++	++	++	+	
Big Data	++		+	++	
Computer Vision	++	++	+	+	
Datenvisualisierung	++	++	+	++	+
Deep Learning	+	++	++	++	
Deep Learning für Computer Vision	+	+	++	+	+
KI für Games - fortgeschrittene Themen	++	++		++	+
Kryptographie	++		+		
Mobile Roboter	++		+	++	
Physically Based Rendering	++	++		+	
Prozedurale Modellierung	+	++	+	+	+
Softwarearchitektur		+	+	+	
Projekt	+	+	+	++	++
Wissenschaftliche Ausarbeitung zu Projekt	++	++		+	++
Masterarbeit	+	++	+	++	+

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.)

Hochschule für angewandte Wissenschaften Kempten

gültig mit Studienbeginn WS 2017/18

Semester	gültig mit Studienbeginn WS 2017/18																														
3	Masterarbeit ⁽⁴⁾																														
2	Computer Vision ⁽¹⁾					Advanced Game Engineering ⁽¹⁾					Augmented Reality ⁽¹⁾					WP-Modul ⁽²⁾					WP-Modul ⁽²⁾					WP-Modul ⁽²⁾					
1	Algorithmen für Real Time Rendering ⁽¹⁾					Simulation und Datenvisualisierung ⁽¹⁾					WP-Modul ⁽²⁾					Wissenschaftliche Ausarbeitung zu Projekt ⁽³⁾					Master-Projekt ⁽³⁾										
	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	ECTS-Punkte
Legende:																															
1		Pflichtmodule										3		Projekt (Modul jeweils aus zwei Teilmodulen)																	
2		Wahlpflichtmodule										4		Masterarbeit																	

Legende:

1

Pflichtmodule

3

Projekt (Modul jeweils aus zwei Teilmodulen)

2

Wahlpflichtmodule

4

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 2017/18

Semester	gültig mit Studienbeginn WS 2017/18																														
6	Masterarbeit ⁽⁴⁾																														
5	Masterarbeit ⁽⁴⁾																														
4	WP-Modul ⁽²⁾					WP-Modul ⁽²⁾					WP-Modul ⁽²⁾																				
3	Wissenschaftliche Ausarbeitung zu Projekt ⁽³⁾					Master-Projekt ⁽³⁾																									
2	Computer Vision ⁽¹⁾					Advanced Game Engineering ⁽¹⁾					Augmented Reality ⁽¹⁾																				
1	Algorithmen für Real Time Rendering ⁽¹⁾					Simulation und Datenvisualisierung ⁽¹⁾					WP-Modul ⁽²⁾																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ECTS-Punkte															
Legende:																															
1	Pflichtmodule															3	Projekt (Modul jeweils aus zwei Teilmodulen)														
2	Wahlpflichtmodule															4	Masterarbeit														

Legende:

1

Pflichtmodule

3

Projekt (Modul jeweils aus zwei Teilmodulen)

2

Wahlpflichtmodule

4

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.

3 Modulbeschreibungen

GEM1101 Algorithmen für Real Time Rendering

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Bernd Dreier
Dozent(en):	Prof. Dr. Bernd Dreier
Modultyp:	Pflichtmodul
Voraussetzungen:	Grundlegende Kenntnisse der OpenGL-Rendering Pipeline entsprechend dem im Bachelor Informatik - Game Engineering angebotenen Fach
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS betreutes Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Leistungsnachweise im Praktikum, Leistungsnachweise sind Zulassungsvoraussetzung schriftl. Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	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 Vulkan und Vulkan Raytracing und können Anwendungen mittlerer Komplexität damit entwickeln
- 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
- Vulkan und Vulkan Raytracing
- Bounding Volumes und ausgewählte Algorithmen der algorithmischen Geometrie

Literatur:

- Eric Lengyel, Mathematics for 3D Game Programming and Computer Graphics, 3rd Edition, 2011
- Akenine-Möller, Real-Time Rendering, AK Peters, 4th edition, 2018
- Computational Geometry, Mark de Berg, et al., Springer, 3rd edition, 2008
- Eric Lengyel, Foundations of Game Engine Development, Vol 2. Rendering, Thera-ton Software, 2019

GEM1102 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.

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

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Teilnahme an der Lehrveranstaltung kennen die Studierenden verschiedene Anwendungsbereiche der wissenschaftlichen 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

Literatur:

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

-
- 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

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

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden sind in der Lage

- den theoretischen Hintergrund klassischer Verfahren und moderner Verfahren zu erklären.
- die gelernten Verfahren zu kombinieren und mittels Pytorch und OpenCV auf Beispiele aus der Praxis zur Objekterkennung und -segmentierung zu übertragen.
- mathematische Zusammenhänge zu verstehen und anzuwenden.

Lehrinhalte:

Folgende Inhalte werden in der Veranstaltung bearbeitet:

- Einführung in Python, OpenCV und Pytorch
- Klassische 2D Computer Vision Verfahren wie z.B. HOG- und SIFT-Detektor
- Basisoperationen für Tiefe Neuronale Netze (Faltung, Max-Pooling, Max-Unpooling, Up-Convolution)
- Convolutional Neural Networks
- Segmentierung (klassisch: Selective Search & modern: U-Net)
- Objektlokalisierung und Objektdetektion
- Moderne Objektdetektoren: R-CNN, Fast-RCNN, Faster R-CNN, YOLO
- Implementierung von klassischen Computer Vision Verfahren mittels OpenCV und modernen Computer Vision Verfahren mittels Pytorch
- Metriken zur Objektdetektion wie z.B. Precision, Recall, F β -Maß, Konfusionsmatrix,

mAP, Multiclass-Confusion-Matrix

Literatur:

Literatur und Empfehlungen zu Einzelthemen werden fortlaufend in der Vorlesung bekanntgegeben

GEM1104 Advanced Game Engineering

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Tobias Breiner
Dozent(en):	Prof. Dr. Tobias Breiner
Modultyp:	Pflichtmodul
Voraussetzungen:	
Verwendbarkeit:	Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	1 SWS Seminaristischer Unterricht 3 SWS Übungen
Leistungspunkte:	5
Arbeitsaufwand:	15 Stunden Präsenzzeit Unterricht 45 Stunden Präsenzzeit Übungen 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftl. Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	handschriftliche Notizen, 1 DIN A4 Blatt, einseitig beschrieben, keine Kopie

Lernergebnisse und Inhalte

Lernergebnisse:

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

- mit Vulkan zu programmieren
- mit DirectX12 zu programmieren
- eine einfache Game-Engine zu erstellen

Lehrinhalte:

- Game Engines, Arten und Einteilung
- Szenegrafen
- Shadersprachen (HLSL, CG und GLSL)
- Object-View-Projection-Model
- Vulkan
- DirectX12
- Aufbau von Mathematikklassen, insb.für homogene Matrizen und Vektoren
- Programmierung einer einfachen Game-Engine
- Renderfenster
- Transfer zur GPU mittels des Upload-Buffer-Konzepts
- Vertex-, Index- und Constant Buffer
- Renderpipeline
- Fences, Ressource Barriers und Semaphores

Literatur:

GEM1105 Augmented Reality

Allgemeines

Modulverantwortliche(r):	Prof. Dr.-Ing. René Bühling
Dozent(en):	Prof. Dr.-Ing. René Bühling
Modultyp:	Pflichtmodul
Voraussetzungen:	hilfreich: Erfahrung mit C#/Unity und Blender
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übungen/Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Prüfungsvorleistung: erfolgreiche Teilnahme an den Übungen. Schriftliche Prüfung von 90 Minuten am Ende des Semesters.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

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 technischen Möglichkeiten eines Augmented Reality Systems zu analysieren.
- geeignete Lösungsansätze für AR-Produkte in einem Anwendungskontext auszuwählen.
- technische Schritte zur Umsetzung einer AR-App einzuschätzen und umzusetzen.
- Usability-Aspekte unter den besonderen Gesichtspunkten der AR zu bewerten und zu optimieren.
- zu beurteilen wann und warum Augmented Reality ein geeignetes Medium für einen Anwendungsfall ist.

Lehrinhalte:

Es sind folgende Themen vorgesehen:

- Überblick über Techniken und Definition von Augmented Reality
- Interaktionsdesign
- Visuelle Ausgabe
- AR-Anwendungen (Outdoor und Indoor)

- Wahrnehmungsaspekte und Informationsvisualisierung
- Gestaltung multimedialer Inhalte
- Gesellschaftliche und ethische Aspekte von Mixed Reality

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>

GEM2102 Deep Learning

Allgemeines

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

- Motivation Deep Learning: kürzliche Erfolge
- Geschichte des Deep Learnings
- Wieso boomt Deep Learning erst jetzt?
- Das biologische Vorbild
- Technische Neuronenmodelle
- Ein Neuronales Netz ohne Deep Learning Bibliothek selber implementieren
- Perzeptron / Perzeptron-Lernregel

- Multi-Layer-Perzeptron
- Backpropagation-Algorithmus
- Deep Learning Bibliotheken
- Deep Learning zur Textverarbeitung
- Optimierer und 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

GEM2103 Algorithmen und Strategien zur Entscheidungsunterstützung

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Jochen Staudacher
Dozent(en):	Prof. Dr. Jochen Staudacher
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	3 SWS Seminaristischer Unterricht 1 SWS Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	45 Stunden Präsenzzeit Vorlesung 15 Stunden Präsenzzeit Praktikum 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Die erfolgreiche Teilnahme an den Übungen gilt als Zulassungsvoraussetzung für eine 20-minütige mündliche Prüfung am Ende des Semesters.
Zur Prüfung zugelassene Hilfsmittel:	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 Liqueur: The R Software, Springer, 2014
- Larry Wasserman: All of statistics, Springer, 2004

GEM2105 Physically Based Rendering

Allgemeines

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

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

- Richards, Mark, Ford, Neil: Handbuch moderner Softwarearchitektur, O'Reilly, 2021
- Starke, Gernot: Effektive Software-Architekturen, Carl Hanser Verlag, 9. Auflage (2020)
- Bass, Len et al.: Software Architecture in Practice, Fourth Edition, SEI Series in Software Engineering, Addison-Wesley (2021)
- Martin, Robert, C.: Clean Architecture, Prentice Hall, 2018
- Richardson, Chris: Microservices Patterns, Manning, 2019
- Oestereich, Bernd: Die UML 2.5 Kurzreferenz für die Praxis, Oldenbourg Verlag, 6. Auflage (2014)
- Buschmann, Frank et al.: POSA Band 1 - Pattern-orientierte Softwarearchitektur - Ein Pattern-System

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.

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

Modulverantwortliche(r):	Prof. Dr. Tobias Breiner
Dozent(en):	Prof. Dr. Tobias Breiner
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	keine
Verwendbarkeit:	Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Sommersemester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht, 30 Stunden Präsenzzeit Übung, 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Benotete Studienarbeit: Elektronische Abgabe eines genau spezifizierten prozedural erstellten Game-Levels.
Zur Prüfung zugelassene Hilfsmittel:	

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

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?

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

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

GEM2110 3D Computer Vision

Allgemeines

Das Modul 3D Computer Vision behandelt ausgewählte Methoden der Projektiven Geometrie, der Posenbestimmung von Objekten und der 3D-Punktwolken-Analyse.

Es werden mathematischen Grundlagen zur Lösung linearer und nichtlinearer Gleichungssysteme behandelt und im Bereich der 3D Computer Vision angewendet. Dabei spielen homogene Koordinaten und homogene Gleichungssysteme eine entscheidende Rolle. Es werden Verfahren zur Kalibration von Kamerasystemen (Single View und Multiple View) behandelt und Methoden zur Triangulation auf der Basis der "Structure From Motion" vorgestellt.

Modulverantwortliche(r):	Prof. Dr. Klaus Ulhaas
Dozent(en):	Prof. Dr. Klaus Ulhaas
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	
Verwendbarkeit:	Studiengang Master Künstliche Intelligenz und Computer Vision, Master Informatik, Master Game-Engineering und Visual Computing
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übungen/Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	nicht programmierbarer Taschenrechner

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden sind in der Lage

- 3D Punkte auf mehrere Kamerabildflächen zu projizieren
- aus Bildpunkten mehrerer Kameras entsprechende 3D Punkte zu rekonstruieren
- ein Mehrkamerasystem zu Kalibrieren
- die Epipolargeometrie zum Matching zu nutzen
- Lineare homogene Gleichungssysteme zu nutzen um unbekannte Posen-, Structure- oder Kameraparameter zu lösen
- metrische Rekonstruktionen durchzuführen

Lehrinhalte:

Das Modul 3D Computer Vision behandelt folgende Themen:

- Projektive Geometrie

- Homographie
- Kamerakalibration
- Epipolargeometrie: Essential und Fundamentalmatrix
- Triangulation und metrische Rekonstruktion
- Lineare und Nichtlineare Optimierung
- Bündelblock-Methoden (Bundle Adjustment)
- Pose Estimation
- Iterative Closest Point (ICP)
- Structure From Motion

Literatur:

Die Literatur wird zu den einzelnen Themen auf den Folien bekanntgegeben. Folgende Bücher sind zu empfehlen:

- 3D Point Cloud Analysis, Liu, Shan, Zhang, Min, Kadam, Pranav and Kuo, C. -C. Jay, Springer International Publishing, 2021
- Multiple View Geometry in Computer Vision, Hartley, Richard and Zisserman, Andrew, Cambridge University Press, Second Edition, 2003

GEM2111 Big Data

Allgemeines

Die Datenmengen, die in unserer Kommunikations- und Wissensgesellschaft anfallen wachsen unaufhörlich und die Datenvielfalt nimmt zu. Sie bieten ein riesiges Potential um mehr über uns, unsere Gesellschaft, unsere Umwelt und unsere Wirtschaft zu erfahren. Daten müssen jedoch zunächst einmal gespeichert, verarbeitet und analysiert werden. Dies stellt ganz neue Anforderungen an die zugrundeliegenden Systeme, da die klassischen Ansätze mit relationalen Datenbanken für strukturierte Daten hierfür nicht mehr geeignet sind. In Big Data Systemen werden die Daten verteilt auf viele Knoten gespeichert und verarbeitet. Die Daten selbst sind oft semistrukturiert oder unstrukturiert und müssen ggf. erst aufbereitet und gefiltert werden. Analysemethoden werden immer ausgefeilter und zunehmend ist eine Analyse in Echtzeit gefordert, die wiederum neue Technologien und Systeme erfordert.

In der Vorlesung zu Big Data werden verschiedene Problemstellungen und Lösungsansätze für Big Data Systeme diskutiert. Im Praktikum werden diese Ansätze mit einigen der derzeit leistungsfähigsten Systeme auf reale Daten angewendet.

Modulverantwortliche(r):	Prof. Nikolaus Steger
Dozent(en):	Prof. Nikolaus Steger
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	jedes Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Vorlesung 30 Stunden Präsenzzeit Praktikum 90 Selbststudium
Leistungsnachweis und Prüfung:	Aktive Teilnahme am Praktikum. Schriftliche Prüfung 90 Minuten am Ende des Semesters. Erfolgreiche Teilnahme am Praktikum ist Zulassungsvoraussetzung für die Prüfung.
Zur Prüfung zugelassene Hilfsmittel:	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage

- die Funktionsweise, die theoretischen Grundlagen und wichtige Algorithmen und Datenstrukturen zur Speicherung, Analyse und verteilten Verarbeitung großer Datenmengen zu verstehen
- konkrete Komponenten für Big Data Systeme zu kennen und in der Praxis einzusetzen
- komplexe Big Data Systeme zu konzipieren und aufzubauen

Lehrinhalte:

- Einsatzbereiche von Big Data Systemen
- Map-Reduce-Paradigma, Apache Hadoop, HDFS
- Batchverarbeitung mit Apache Spark, Resilient Distributed Datasets, Data Frames und Spark SQL
- Data Warehousing
- SQL-Erweiterungen für Analytics und Data Warehousing
- Column Store, Datenkomprimierung und Datenstrukturen zur Speicherung analytischer Daten
- Online Analytics und Stream Processing mit Apache Flink und Flink SQL
- Anforderungen, allgemeine Konzepte und Komponenten für Big Data Systeme

Literatur:

- Learning Spark, Lightning-Fast Data Analysis- Damji, Wenig, Das, O'Reilly and Associates

GEM2112 Deep Learning für Computer Vision

Allgemeines

In diesem Modul werden Deep Learning Modelle und Techniken für die Verarbeitung von Bildern vermittelt. Im Fokus hierbei steht das Convolutional Neural Network (CNN), aber auch Varianten und neuere Ansätze werden vorgestellt. CNNs stellen heute den "Kern" vieler State-of-the-Art (SOTA) Computer Vision Algorithmen dar.

Wir starten das Modul mit einer kurzen Geschichte der Bildverarbeitung. Die Herausforderungen der Bildverarbeitung und wie diese über viele Jahrzehnte hinweg systematisch unterschätzt worden sind, werden aufgeführt. Danach wird das CNN eingeführt, das heute das "Arbeitspferd" der Deep Learning basierten Bildverarbeitung darstellt und sehr oft als sog. "Feature Extractor" oder "Backbone" für viele neuere Bildverarbeitungsmodelle verwendet wird. Um das CNN Basismodell richtig zu durchdringen werden wir die einzelnen Komponenten eines CNNs (Filterschichten, Poolingschichten, Klassifikatorschicht) genauer untersuchen und die Studierenden sollen anhand von Experimenten in den Übungen mit Varianten dieser Schichten arbeiten, um eine Intuition für die Bedeutung im Rahmen der gesamten Verarbeitung eines Bildes zu entwickeln.

Es wird dann im Modul auf weiterführende Modelle eingegangen, die zur Objektdetektion in Bildern entwickelt wurden. Ebenso werden Modelle vorgestellt, die es erlauben, Objekte in Bildern pixelgenau zu segmentieren. Neuere Tracking-Verfahren, die Deep Learning Techniken für das Verfolgen von Objekten in Bildfolgen einsetzen sind ein weiterer Baustein dieses Moduls. Abschließend betrachten wir auch Verfahren, um realistisch aussehende künstliche Bilder zu erzeugen und Bilder aus einer Domäne in eine andere Domäne zu transferieren (Image-to-Image bzw. Video-to-Video Translation).

Modulverantwortliche(r):	Prof. Dr. Jürgen Brauer
Dozent(en):	Prof. Dr. Jürgen Brauer
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Das Modul Deep Learning ist explizit keine (!) Voraussetzung
Verwendbarkeit:	
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Leistungsnachweise in den Übungen. Leistungsnachweise sind Zulassungsvoraussetzungen zur schriftlichen Prüfung. Schriftliche Prüfung 90 Minuten am Ende des Semesters.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden kennen die Herausforderungen bei der Bildverarbeitung, können diese durch konkrete Beispiele benennen.
- Die Teilnehmer des Moduls haben einen Überblick über die Geschichte der Bildverarbeitung bis hin zu Deep Learning basierten Modellen der Bildverarbeitung.
- Die Studierenden verstehen die Bedeutung der einzelnen Schichten eines Convolutional Neural Networks und wie die verschiedenen Schichtenarten bei der Bildklassifikation zusammenspielen.
- Die Teilnehmer des Moduls kennen aktuelle Modelle zur Objektdetektion, zur Segmentierung und zur Objektverfolgung und wissen wie diese sich unterscheiden. Sie verstehen die interne Arbeitsweise und können Vor- und Nachteile der einzelnen Ansätze benennen.
- Die Studierenden kennen Modelle zur Erzeugung realistisch aussehender Bilder und können deren Funktionsweise nachvollziehen.
- Die Teilnehmer des Moduls haben einen Überblick über Image-to-Image- und Video-to-Video-Translation-Ansätzen
- Die Studierenden können einzelne ausgewählte Deep Learning Verfahren zur Bildverarbeitung mit aktuellen Deep Learning Bibliotheken umsetzen.

Lehrinhalte:

- Computer Vision - Eine kurzer geschichtlicher Rückblick
- CNN - Gesamtüberblick über das Convolutional Neural Network
- CNN - Bedeutung und Funktionsweise der einzelnen Komponenten
- Deep Learning Modelle zur Objektdetektion
- Deep Learning Modelle zur Segmentierung
- Deep Tracking Verfahren zur Objektverfolgung
- Generative Deep Learning Modelle um realistisch aussehende Bilder zu erzeugen
- Image-to-Image- und Video-to-Video Translation

Literatur:

GEM2114 Künstliche Intelligenz für Games - fortgeschrittene Themen

Allgemeines

Die Vorlesung baut auf die Themen des WPM „Künstliche Intelligenz für Games“ im Bachelorstudiengang Game Engineering auf. Sie kann aber auch ohne Vorkenntnisse besucht werden. Inhaltlich werden folgende Themen behandelt: Wiederholung der Standardalgorithmen zur Bewegung und Entscheidungsfindung, Wiederholung der Architektur einer KI Engine, Bewegung von großen NPC-Gruppen, Planungsalgorithmen, taktische und strategische Techniken, Ballistik und lernende Algorithmen.

Modulverantwortliche(r):	Prof. Dr. Christoph Bichlmeier
Dozent(en):	Prof. Dr. Christoph Bichlmeier
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Kenntnisse in C++ und OOP
Verwendbarkeit:	
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übungen (verpflichtend)
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	schriftl. Prüfung 90 Minuten am Ende des Semesters Teilnahmepflicht im Praktikum, Leistungsnachweise im Praktikum Leistungsnachweise sind Zulassungsvoraussetzungen
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Teilnahme an der Lehrveranstaltung können die Studierenden ...

- per Algorithmen große Gruppen von NPCs durch komplexes Terrain navigieren
- eine Auswahl von Planungsalgorithmen implementieren
- taktische und strategische Algorithmen bewerten und umsetzen
- eine Auswahl von lernenden Algorithmen implementieren und bewerten

Lehrinhalte:

Im Rahmen der Veranstaltung ...

- erhalten die Studierenden einen umfassenden Blick auf die Techniken der Verhaltenssteuerung von NPCs
- setzen die Studierenden eine Auswahl von Algorithmen praktisch in CPP um
- lernen die Studierenden Konzepte zur Navigation von großen NPC Gruppen kennen

- wird vermittelt wie ballistische Information zu Entscheidungsfindung genutzt werden kann
- werden verschiedene taktische Karten, deren Implementierung und Kombination erklärt
- wird das Thema Planung als Variante der Entscheidungsfindung praktisch vermittelt
- wird eine Auswahl an lernenden Algorithmen behandelt, die in Computerspielen beispielsweise für die Entscheidungsfindung zum Einsatz kommen können.

Literatur:

Folgende Literatur wird empfohlen:

- Künstliche Intelligenz (Pearson Studium - IT) von Stuart Russell, Peter Norvig
- Artificial Intelligence for Games von Ian Millington, John Funge
- Game AI Pro 1-3: Collected Wisdom of Game AI Professionals
- Weitere themenbezogene wiss. Paper

GEM5100.1 Projektarbeit

Allgemeines

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

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

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

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

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: