

LEHRSTUHL FÜR RECHNERARCHITEKTUR UND PARALLELE SYSTEME

**Aspekte der systemnahen Programmierung  
bei der Spieleentwicklung****Praktikumsordnung**  
(Stand: 11. Juni 2018)**Inhaltsverzeichnis**

<b>1 Grundsätzliches</b>	<b>1</b>
<b>2 Durchführungsform</b>	<b>2</b>
2.1 Form des Praktikums . . . . .	2
2.2 Vorgehensweise . . . . .	2
2.3 Dauer der Bearbeitung . . . . .	2
2.4 Verhalten der Gruppenmitglieder . . . . .	2
<b>3 Anforderungen</b>	<b>2</b>
3.1 Ausarbeitung . . . . .	3
3.2 Implementierung . . . . .	4
3.3 Projektpräsentation . . . . .	4
<b>4 Bewertung</b>	<b>5</b>
4.1 Notenbildung . . . . .	5
4.2 Notenvergabe . . . . .	5
4.3 Einsicht . . . . .	5
<b>5 Sonderfälle</b>	<b>6</b>
5.1 Gruppen aus weniger als drei Personen . . . . .	6
5.2 Abwesenheit am Vortragstermin . . . . .	6
<b>6 Möglichkeiten der Sonderprüfung</b>	<b>6</b>
<b>7 Schlussbemerkung</b>	<b>6</b>
<b>A1 Bewertungskriterien</b>	<b>7</b>

**1 Grundsätzliches**

Diese Praktikumsordnung soll einen ordnungsgemäßen Ablauf des Praktikums „Aspekte der systemnahen Programmierung bei der Spieleentwicklung“ (ASP) gewährleisten. Sie ist nicht Bestandteil der Prüfungsordnung und gibt auch keine Hinweise auf den fachlichen Inhalt des Praktikums.

## 2 Durchführungsform

### 2.1 Form des Praktikums

Das Praktikum wird in Form von Gruppenprojekten durchgeführt. In Vorbereitung auf die Projekte werden in der ersten Hälfte des Semesters im Praktikumsbetrieb und durch freiwillige Hausaufgaben die notwendigen Grundlagen vermittelt. Die hier erarbeiteten Kenntnisse werden für die Projektaufgaben vorausgesetzt. Im zweiten Teil des Semesters bekommt jede Gruppe (bestehend aus 3 Personen) ein Projekt zugeteilt, welches in einer vorgegebenen Zeit zu bearbeiten ist. Für das erfolgreiche Bestehen des Praktikums sind eine Ausarbeitung, die Projektimplementierung und die Projektpräsentation Voraussetzung. Die Ausarbeitung und die Projektpräsentationen sind in *deutscher Sprache* anzufertigen bzw. vorzutragen.

### 2.2 Vorgehensweise

Die (gewollt) offen gestellten Aufgaben sind in der ersten Bearbeitungswoche zu konkretisieren und mit dem Tutor zu besprechen. Es wird erwartet, dass jedes Gruppenmitglied an allen Teilen mitarbeitet und insbesondere zu jedem Teil Fragen beantworten kann.

### 2.3 Dauer der Bearbeitung

Das Praktikum ist in einem Semester zu absolvieren. Mit Ausgabe der Projektaufgaben wird ein Abgabetermin festgelegt, zu welchem das Ergebnis eingereicht werden muss. Sollte diese Frist nicht eingehalten werden können, ist ein Ausnahmeantrag zu stellen. Bei diesem entscheidet die Praktikumsverwaltung in Rücksprache mit dem Tutor, ob eine Verlängerung möglich ist. Eine zweite Verlängerung der Frist ist nicht möglich. Sollte der Ausnahmeantrag nicht genehmigt und das Projekt trotzdem erst nach Fristende fertiggestellt werden, gilt das Praktikum als nicht bestanden.

### 2.4 Verhalten der Gruppenmitglieder

Die Gruppenmitglieder haben nachzuweisen, dass sie sich um die Integration aller Gruppenmitglieder bemüht haben. Sollten dennoch Gruppen mit weniger als drei Personen zustande kommen, so gelten die Ausnahmeordnungen unter Punkt 5.1.

## 3 Anforderungen

Das Praktikums-Projekt besteht aus 3 Teilen, welche hier näher beschrieben werden: Ausarbeitung, Implementierung und Projektpräsentation. Die Projekte werden über ein Versionsverwaltungs-System (git) für die Gruppen bereitgestellt und sind auch über dieses abzugeben. Die Zuordnung erfolgt durch den Lehrstuhl für Rechnerarchitektur und Parallele Systeme.

### 3.1 Ausarbeitung

Über die Bearbeitung des Projekts ist eine 9-11 seitige Ausarbeitung anzufertigen. Folgende Bereiche sind zu behandeln. Der Umfang der einzelnen Bereiche ist sinnvoll zu wählen.

1. *Einleitung*  
Kurze, prägnante Beschreibung des Praktikums und der Kern-Inhalte.
2. *Problemstellung und Spezifikation*  
In diesem einleitenden Abschnitt muss die Aufgabenstellung analysiert und gegebenenfalls näher spezifiziert werden. Dies sollte in Absprache mit dem betreuenden Tutor geschehen.
3. *Lösungsfindung*  
Der Weg von der Aufgabenstellung zu der gewählten und umgesetzten Implementierung ist zu beschreiben. Ein klar erkennbarer Entscheidungsprozess beschreibt, welche Aspekte bei der Lösungsfindung diskutiert wurden, und was bei der Umsetzung speziell beachtet wurde. Wo möglich beinhaltet dies eine Gegenüberstellung möglicher Lösungsalternativen und Überlegungen zu möglichen Optimierungen.
4. *Dokumentation der Implementierung*  
In diesem Abschnitt wird das erstellte Projekt in seiner Gesamtheit beschrieben. Es dient als Dokumentation der Implementierung. Sowohl Benutzer-Dokumentation als auch Entwickler-Dokumentation müssen erstellt werden.  
Zusammen mit der gut kommentierten Implementierung soll diese Dokumentation es ermöglichen, das erstellte Projekt zu verwenden und die Implementierung leicht nachvollziehen zu können. Vor allem sind die technischen Details der gewählten Optimierungen und Code-Strukturierungen hier zu beschreiben.
5. *Ergebnisse*  
Im letzten Teil der Ausarbeitung soll die Implementierung analysiert und bewertet werden. Hierzu sind geeignete Methoden zu wählen, die unter anderem Zeitmessungen, Vergleiche zum Disassembly-Code vergleichbarer Implementierungen in C und Analysen über die Korrektheit und Genauigkeit der Ergebnisse beinhalten sollten.
6. *Zusammenfassung und Ausblick*  
Neben einer kurzen Zusammenfassung des umgesetzten Projektes sollte dieser Abschnitt einen kurzen Ausblick enthalten. In diesem wird beurteilt, wo zusätzliches Potential für Verbesserungen existiert, ob rückblickend eine andere Lösungsalternative besser wäre, und wie die Qualität der eigenen Implementierung eingeschätzt wird.
7. *Quellenverzeichnis*  
Sämtliche in der Ausarbeitung verwendeten Quellen sind hier aufzuführen.

Die Ausarbeitung ist auf Basis der L<sup>A</sup>T<sub>E</sub>X-Vorlage zu erstellen, die im Rahmen des Praktikums bereitgestellt wird. Abzugeben sind neben einer PDF-Version die L<sup>A</sup>T<sub>E</sub>X-Quellen und alle weitere Dateien, die zum Erzeugen des PDFs notwendig sind.

### 3.2 Implementierung

Der aus der Aufgabenstellung erarbeitete Algorithmus muss wie in den einführenden Übungen in Assembler implementiert werden. Das Rahmenprogramm ist so zu erstellen, dass neben evtl. verwendeten Zeitmessungen auch unterschiedliche, repräsentative Aufrufe in den implementierten Algorithmus getätigt werden. Eine sinnvolle Ausgabe in Bezug auf Testfälle und Zeitmessung wird erwartet. Im Rahmenprogramm können auch Referenzimplementierungen eingefügt werden, mit denen die eigene Implementierung verglichen wird.

Sehr wichtig ist das ausführliche und sinnvolle Kommentieren des Codes (sowohl Assembler als auch Rahmenprogramm)! Keine oder nichtssagende Kommentare werden die Benotung negativ beeinflussen. In Bezug auf die eigentliche Assembler-Implementierung ist zu beachten, dass die Lösung zu optimieren ist, also möglichst alle unnötigen Befehle/Schritte weggelassen werden.

Bei Aufgaben, bei denen kein Assembler-Code geschrieben wird (beispielsweise Reverse-Engineering Aufgaben), ist all derjenige Code einzureichen, der zur Umsetzung entstanden ist (z.B. Analyse-Code). In diesem Fall wird der Inhalt der Ausarbeitung gleichermaßen als Implementierung bewertet.

### 3.3 Projektpräsentation

Die Projektpräsentation ist zweigeteilt.

Kurz nach der Vergabe der Aufgaben an die Projektgruppen ist im Rahmen der Tutorübungen eine kurze Zusammenfassung der Aufgabe zu präsentieren. Diese ist auf 2 Minuten beschränkt und wird von einem Gruppenmitglied vorgestellt. Die Anwesenheit aller Gruppenmitglieder ist verpflichtend, es ist ein Tutorium zu wählen, in der mindestens ein Gruppenmitglied angemeldet ist.

Im Anschluss an das Praktikum muss jede Gruppe einen Projektvortrag von 15 Minuten halten (hartes Zeitlimit!). In diesem sollen die wichtigen Aspekte des Projekts vorgestellt werden. Eine sinnvolle Wahl der Inhalte ist Teil der Aufgabenstellung. Es wird erwartet, dass der Vortrag in ungefähr gleichen Teilen von allen Gruppenmitgliedern gehalten wird. Die Folien der Abschlusspräsentation sind zusammen mit den anderen Dokumenten *am Abgabetermin des Projekts im PDF-Format* abzugeben.

Im Rahmen des Vortrags können jedem Gruppenmitglied zu jedem Teil des Projekts Fragen gestellt werden. Zusätzlich werden die Praktikumsleitung und die Tutoren die Möglichkeit haben, den Teilnehmern Fragen zu den im Praktikum erlernten Inhalten, den verwendeten Hardware- und Software-Komponenten und der relevanten Theorie zu stellen. Die Fragen zur Theorie basieren auf den einführenden Stunden der Vorlesung „Betriebssysteme und hardwarenahe Programmierung für Games“ (IN0034).

## 4 Bewertung

Es werden nur Studierende benotet, deren Gruppe die Ausarbeitung, die Präsentation und den Projekt-Code eingereicht und beide Teile der Projektpräsentation gehalten haben. Die Tutoren und Projektbetreuer bewerten die Ergebnisse nach den in den Bewertungsbögen aufgeführten Kriterien.

### 4.1 Notenbildung

Aus den Bewertungskriterien ergeben sich Einzelbewertungen für Ausarbeitung, Implementierung und Vortrag, für welche jeweils 20 Punkte erzielt werden können. Es müssen für *jede* dieser Einzelbewertungen mindestens 6.5 Punkte erreicht werden, andernfalls wird das Praktikum direkt mit der Note 5.0 bewertet. Die Gesamtpunktzahl ergibt sich aus der Summe dieser Einzelbewertungen, sodass insgesamt 60 Punkte erreicht werden können. Die Note berechnet sich anhand von Tabelle 1.

Punkte $\geq$	58.5	55.5	52.5	46.5	43.5	40.5	37.5	34.5	31.5	25.5	22.5	19.5	< 19.5
Note	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7	4.0	4.3	4.7	5.0

Tabelle 1: Berechnung der Noten

### 4.2 Notenvergabe

Die Noten werden ausschließlich über TUMonline vergeben. Zum Bestehen des Praktikums muss eine Anmeldung in TUMonline zur Prüfung im entsprechenden Semester vorliegen. Bei nicht fristgerechter Anmeldung zur Prüfung auf TUMonline gilt das Praktikum als nicht bestanden.

Sonderfälle sind Studierende, die keine TUMonline-Kennung besitzen, wie z.B. Schülerstudenten. In diesen Fällen wird auch kein Schein über das Praktikum erstellt, sondern eine Bestätigung über das Bestehen inklusive der Note ausgehändigt.

### 4.3 Einsicht

Spätestens mit Freigabe der Noten wird ein Termin bekannt gegeben, bei dem sich die Studierenden über die Zusammensetzung der Note informieren können (Einsicht). Jede Gruppe kann insgesamt einen Einsichtstermin bekommen. Auf Wunsch der Studierenden kann die Bewertung der Vortragsleistung auch in Abwesenheit der anderen Gruppenmitglieder bekannt gegeben werden. Ein Einsichtstermin ist pro Gruppe auf 15 Minuten beschränkt. Beanstandungen sind am Einsichtstag schriftlich (nicht per E-Mail) bei der Praktikumsleitung einzureichen und werden im Anschluss überprüft. Über die Beanstandung soll innerhalb von zwei Wochen beschieden und der Studierende per E-Mail informiert werden. Durch eine etwaige Nachkorrektur kann es auch zu einer Verschlechterung der Bewertung kommen.

## **5 Sonderfälle**

### **5.1 Gruppen aus weniger als drei Personen**

Sollte eine Gruppe nur aus zwei Personen bestehen, so ist nach Absprache eine entsprechend kürzere Ausarbeitung anzufertigen. Außerdem verkürzt sich die Vortragsdauer auf 10 Minuten.

Bei der Gruppenbildung ist darauf zu achten, dass keine Einzelpersonen übrig bleiben und es maximal viele Dreiergruppen gibt. D.h., eine Einzelperson und eine Dreiergruppe teilen sich zusammen in zwei Zweiergruppen. Sollte von einer Gruppe nur eine Person übrig bleiben, und dies aus Gründen, die diese Person nicht zu vertreten hat, so ist der Projektaufwand in Absprache mit den Tutoren und der Praktikumsleitung weiter zu reduzieren.

### **5.2 Abwesenheit am Vortragstermin**

Sollte ein Gruppenmitglied am bekannt gegebenen Vortragstermin nicht anwesend sein, so wird von den übrigen Gruppenmitgliedern der Vortrag in normaler Länge gehalten. Bei Abwesenheit aus einem anerkannten Grund (z.B. nachgewiesene Krankheit) findet innerhalb von einem Monat, spätestens jedoch in der ersten Vorlesungswoche des folgenden Semesters, eine Einzelprüfung statt. Bei unbegründeter Abwesenheit gilt das Praktikum für den abwesenden Studierenden als nicht bestanden.

## **6 Möglichkeiten der Sonderprüfung**

Die Praktikumsleitung kann bei Verdacht auf ein Plagiat eine mündliche Prüfung der kompletten Gruppe über das Projekt durchführen. Diese mündliche Prüfung beeinflusst maßgeblich die Note bzw. das Bestehen des Praktikums.

Bei bestätigten Plagiaten wird das Praktikum automatisch mit 5.0 bewertet und gilt als nicht bestanden. Abgegebener Assemblercode, der von einem Compiler aus einer Hochsprache generiert wurde, wird ebenfalls als Plagiat gewertet.

## **7 Schlussbemerkung**

In allen hier nicht aufgeführten Fällen entscheidet die Praktikumsleitung in Absprache mit Herrn Prof. Dr. Schulz über den Erhalt des Scheins. Dies gilt insbesondere, falls nach Meinung des Studierenden ein begründeter Härtefall vorliegen sollte. In diesem Fall kann der Studierende dies bei der Praktikumsleitung melden und einen entsprechenden Antrag stellen.

Garching, den 11. Juni 2018  
Prof. Dr. Martin Schulz

## A1 Bewertungskriterien

Folgende Tabellen stellen eine Übersicht über die *ungefähre* Verteilung der Punkte der zu bewertenden Projektbestandteile dar. In begründeten Fällen sind Abweichungen von diesem Schema möglich.

### A1.1 Ausarbeitung

Problemstellung & Spezifikation (4 Punkte)	Lösungsfindung (4 Punkte)	Dokumentation d. Implementierung (4 Punkte)	Ergebnisse (4 Punkte)	Äußere Form (4 Punkte)
Siehe Abschnitt 3.1				

### A1.2 Implementierung

Funktionalität & Richtigkeit (4 Punkte)	Art der Implementierung (4 Punkte)	Dokumentation (4 Punkte)	Tests (4 Punkte)	Äußere Form (4 Punkte)
<ul style="list-style-type: none"> <li>• Abgabe kompiliert, läuft durch</li> <li>• Berechnung erwarteter Ergebnisse</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Kein redundanter Code</li> <li>• Code optimiert</li> <li>• Calling Convention eingehalten</li> <li>• Keine Segmentation Faults</li> <li>• Korrekte Speicherverwaltung</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Gute und sinnvolle Kommentare</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Rahmenprog. mit sinnvollen Testfällen</li> <li>• Tests auf Korrektheit und Performanz</li> <li>• Erkennbare Teststrategie (in Ausarbeitung beschrieben)</li> <li>• Übersichtliche Ausgabe von Ergebnissen</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Codestruktur</li> <li>• Einrückungen</li> <li>• Generelle Lesbarkeit</li> <li>• Passende Variablennamen</li> <li>• etc.</li> </ul>

### A1.3 Vortrag

Problemstellung & Spezifikation (4 Punkte)	Darstellung der Lösungsfindung (4 Punkte)	Foliengestaltung (4 Punkte)	Vortragsstil & Auftreten (4 Punkte)	Fachliches Verständnis (4 Punkte)
<ul style="list-style-type: none"> <li>• Einführung in Thema und Aufgabenstellung</li> </ul>	<ul style="list-style-type: none"> <li>• Lösungsmöglichkeiten</li> <li>• Optimierungen</li> <li>• Schwierigkeiten</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Übersichtlichkeit</li> <li>• „Roter Faden“</li> <li>• Lesbarkeit</li> <li>• Foliendesign</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Auftreten der Vortragenden</li> <li>• Mediennutzung</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Bewertung Fragen zu Projektdetails, Theorie, Hardware, Software, etc.</li> </ul>