

SPIEL – Software-Programmierung intensiv erleben und lernen

Alison McNamara, Jonathan Klemm, Laurin Niclas Dörre

Hochschule	Technische Hochschule Aschaffenburg
Fachbereich	Ingenieurwissenschaft und Informatik (IWIN)
Projektname	SPIEL – Software-Programmierung intensiv erleben und lernen
Teammitglieder	· Prof. Dr. Alison McNamara – Professorin für mobile Anwendungen · Jonathan Klemm – Mitarbeiter Innovative Lehre · Laurin Niclas Dörre – Student im Master Angewandte Forschung in den Ingenieurwissenschaften
Zielgruppe des Projekts	Grundlagen der Programmiertechnik, 1. Semester, ca. 100 Studierende
Projektziele	· positive Lernerfahrung und Einstellung gegenüber dem Fachgebiet fördern · sicherstellen, dass alle Studierenden die Lernziele erreichen · das individuelle Wachstum eines jeden Studierenden fördern
Zentrale Misfits	· »Erfolg hängt zu stark von den Fähigkeiten ab« · »Spiel ist zu leicht zu gewinnen« · »Gegenseitige Beeinflussung ist zu stark«

Zentrale Spielelemente	· »Feedback« · »Spieler-Fortschrittsanzeige« · »Aussetzen«
-------------------------------	--

Schlagworte: *Heterogenität, Game-based Learning, Programmieren, Gamification, Binnendifferenzierung*

Das Projekt SPIEL – Software-Programmierung Intensiv erleben und lernen wurde an der Technischen Hochschule Aschaffenburg entwickelt, um Studierende im ersten Semester des Software-Design-Bachelorprogramms spielerisch an die Grundlagen der Programmierung heranzuführen. Das Pflichtmodul »Grundlagen der Programmiertechnik« bietet eine Einführung in Konzepte der Programmierung und setzt Java als primäre Sprache ein.

Es gibt jedoch zwei zentrale Hürden: Zum einen stellt die Heterogenität der Studierenden, insbesondere in Bezug auf deren Vorkenntnisse, eine große Herausforderung dar. Zum anderen erfordert das Programmieren eine hohe Abstraktionsleistung und logisches Denken, was viele Studierende überfordert.

1. Ausgangslage und Zielsetzung

Das Pflichtmodul »Grundlagen der Programmiertechnik« richtet sich an Studierende im ersten Semester des Bachelorstudiengangs Software Design an der Technischen Hochschule Aschaffenburg. Das Modul bietet eine Einführung in die grundlegenden Konzepte und Techniken der Programmierung und ist speziell auf den Studiengang Software Design ausgerichtet. Es deckt wichtige Themen wie die Syntax und Semantik von Programmiersprachen, Objektorientierung und die Implementierung einfacher Algorithmen ab, wobei der Fokus auf der Programmiersprache Java liegt. Ziel des Moduls ist es, den Studierenden nicht nur technische Fähigkeiten zu vermitteln, sondern auch ein tiefes Verständnis für die Rolle der Informatik im Rahmen der Digitalisierung und des digitalen Wandels zu entwickeln. Es bestehen allerdings zwei zentrale

Ausgangsprobleme innerhalb des Moduls, die mithilfe der EMPAMOS-Methode analysiert und gelöst werden sollen.

Zum einen wäre da die Heterogenität der Studierenden an Hochschulen, die sich in verschiedenen Dimensionen widerspiegelt und eine signifikante Herausforderung für die Konzeption der Lehre insgesamt darstellt (Voit et al., 2022). Soziodemografische Merkmale, Bildungshintergrund, Zugangswege zur Hochschule und das verfügbare Zeitbudget für das Studium sind entscheidende Faktoren, die die Voraussetzungen und Leistungen der Studierenden beeinflussen (Feldhaus & Speck, 2020). In der Lehrveranstaltung ist diese Heterogenität der Studierenden hinsichtlich ihrer Vorkenntnisse und technischen Ressourcen sehr stark ausgeprägt. Einige Teilnehmende bringen eine einschlägige Ausbildung im IT-Bereich mit, während andere, primär durch die positiven Berufsaussichten motiviert, ohne eigenen Computer das Studium beginnen. Dies ist didaktisch sehr herausfordernd und führt immer wieder zu Frustration, Überforderung und negativen Erfahrungen auf Seiten der Studierenden.

Neben der studentischen Heterogenität stellt zum anderen aber auch der hohe Abstraktionsgrad der Programmiersprachen eine zentrale Herausforderung dar. Jede Programmiersprache hat ihre eigene Syntax und Semantik, also Regeln, wie Codes geschrieben und interpretiert werden. Hinzu kommt, dass Programmieren ein hohes Maß an abstraktem und logischem Denken verlangt. Die Studierenden müssen daher lernen, komplexe Probleme in kleinere Teile zu zerlegen und diese systematisch zu lösen. Diese Art des Denkens ist für viele am Anfang nicht intuitiv, was den Lernprozess erschwert (Amnouchokanant et al., 2021).

Diese Kombination aus neuen Sprachen, abstraktem Denken und der Anwendung technologischer Tools macht das Programmieren zu einer komplexen und anspruchsvollen Disziplin. Vor diesem Hintergrund hat sich unser Team zum Ziel gesetzt,

- eine positive Lernerfahrung und Einstellung gegenüber dem Fachgebiet zu fördern,
- sicherzustellen, dass alle Studierenden die Lernziele erreichen, und
- das individuelle Wachstum aller Studierenden zu fördern.

2. Misfit-Analyse mit EMPAMOS

Zu Beginn des Projekts haben wir mit der EMPAMOS-Methode die sogenannten *Misfits* identifiziert, d.h. diejenigen Elemente des Moduls, die sich negativ auf die Motivation der Lernenden auswirken können. Wir konnten mit der Methode eine neue Perspektive auf die Probleme einnehmen und darauf basierend passende Lösungsansätze entwickeln. Im Folgenden werden die zentralen Misfits des Moduls beschrieben.

Abbildung 1: Die Misfit-Karte »Erfolg hängt zu stark von den Fähigkeiten ab«



Voit, T.; Bildrechte ICONS siehe Anhang

Das Misfit »*Erfolg hängt zu stark von den Fähigkeiten ab*« beschreibt, dass bereits in den ersten Veranstaltungen große Diskrepanzen in den Kompetenzen der Studierenden auftreten. Dies führt zu ungleichen Erfolgchancen, da einige Studierende das Modul meistern, während andere noch Schwierigkeiten

haben, überhaupt den Einstieg zu finden. Das Misfit »Erfolg hängt zu stark von den Fähigkeiten ab« beschreibt somit ein zentrales Problem des Moduls. Bereits in den ersten Veranstaltungen wird die große Diskrepanz zwischen den Fähigkeiten und Kompetenzen der Studierenden sichtbar. Diese unterschiedlichen Voraussetzungen haben direkte Auswirkungen auf die Erfolgsaussichten im Modul. Während einige Studierende längst über fundierte Programmierkenntnisse verfügen und die Lernziele des Moduls schon erreicht haben, verfügen andere Studierende nur über begrenzten Zugang zu Computertechnologien und haben infolgedessen bereits zu Beginn des Moduls Schwierigkeiten mit den Grundlagen.

Abbildung 2: Die Misfit-Karte »Wettbewerb ist zu stark«



Voit, T.; Bildrechte ICONS siehe Anhang

Durch die Anwesenheit erfahrener Studierender, die den Lernstoff als zu einfach empfinden, entsteht eine starke Wettbewerbssituation. Dies führt dazu, dass unerfahrene Studierende sich entmutigt fühlen und ihre Teilnahme am Kurs vermindern. Erfahrene und unerfahrene Studierende stehen über das gesamte Semester hinweg außerdem in einem Wettstreit um die Aufmerksamkeit und Zeit der Professorin. Die erfahrenen Studierenden empfinden den durchgenommenen Lernstoff als nicht herausfordernd oder gar langweilig, was zu störendem Verhalten in der Vorlesung führen kann. Diese Studierenden lenken ihre Kommiliton:innen ab und ziehen die Aufmerksamkeit der Professorin durch fortgeschrittene Fragen auf sich, die nicht zum aktuellen Lehrplan passen. Infolgedessen bleibt weniger Zeit für Erklärungen und Beispiele, die den weniger erfahrenen Studierenden helfen könnten.

Abbildung 3: Die Misfit-Karte »Gegenseitige Beeinflussung zu stark«



Voit, T.; Bildrechte ICONS siehe Anhang

Die fortgeschrittenen Studierenden tragen oft unabsichtlich zur Verunsicherung ihrer Kommiliton:innen bei, indem sie ihre Fähigkeiten zur Schau stellen, was bei weniger erfahrenen Studierenden Gefühle der Unterlegenheit hervorruft. Erfahrene Studierende können außerdem – beabsichtigt oder unbeabsichtigt – eine wettbewerbsorientierte Atmosphäre schaffen, indem sie ihre Fähigkeiten zur Schau stellen. Dieses Verhalten schüchtert unerfahrene Studierende zusätzlich ein, was zu Gefühlen der Unzulänglichkeit und einem verringerten Selbstwertgefühl führt. Infolgedessen beteiligen sich weniger erfahrene Studierende weniger aktiv am Unterricht, was ihre Chancen auf Lernerfolge weiter verringert. Die wahrgenommene Kluft in den Fähigkeiten kann zudem ein Gefühl der Ausgrenzung hervorrufen und somit die aktive Teilnahme der weniger erfahrenen Studierenden hemmen. Dies führt letztendlich dazu, dass die unerfahrenen Studierenden das Programmieren überwiegend negativ konnotieren, kein Interesse oder gar Freude für das Fachgebiet entwickeln und somit ein sich selbst verstärkender negativer Kreislauf beginnt – denn Studien belegen, dass die Einstellung der Studierenden zum Programmieren einen erheblichen Einfluss auf ihre Leistungen hat (Amnouchokanant et al., 2021).

Erfahrene Studierende haben wenig Motivation, das Modul zu durchlaufen, weil sie es als zu leicht empfinden. Das schränkt ihre aktive Teilnahme ein. »*Spiel ist zu leicht zu gewinnen*« ist daher ein weiteres, bereits thematisiertes Misfit des Moduls. Es ist mit den anderen Misfits eng verwoben, denn sobald erfahrene Studierende das Modul nicht als Herausforderung erleben, steigt auch das Risiko, dass Verhaltensweisen einsetzen, aus denen die bereits dargestellten Misfits »*Wettbewerb ist zu stark*« und »*Gegenseitige Beeinflussung ist zu stark*« resultieren. Darüber hinaus stellt die Unterforderung mancher Studierenden auch für sich allein genommen eine Hürde für das erfolgreiche Lernen dar. Ergebnisse aus früheren Semestern zeigen, dass etwa 40 % der Studierenden die Lerninhalte des Moduls zu einem großen Teil bereits beherrschen und das Gefühl haben, nichts Neues dazulernen zu können. Diese Zahl basiert auf einer internen Auswertung von Klausurergebnissen und Studierendenbefragungen, die regelmäßig durchgeführt werden, um den Fortschritt zu evaluieren.

Abbildung 4: Die Misfit-Karte »Spiel ist zu leicht zu gewinnen«



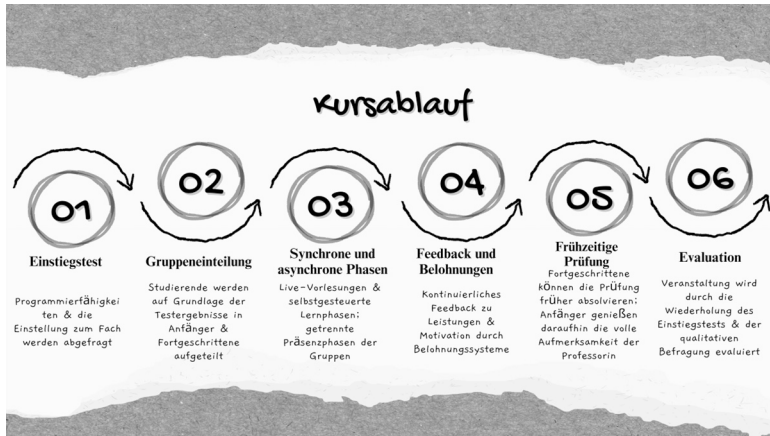
Voit, T.; Bildrechte ICONS siehe Anhang

3. Lösungsansätze

Die ermittelten Misfits unterstreichen die Bedeutung differenzierter Instruktionen innerhalb des Kurses »Grundlagen der Programmietechnik«. Ohne angemessene Differenzierung bzw. *Binnendifferenzierung* versäumt es der Kurs, den unterschiedlichen Fähigkeitsniveaus der Studierenden gerecht zu werden, was zu einer insgesamt suboptimalen Lernumgebung führt. Das Lehrprojekt identifizierte daher mittels der »Spielemente« der EMPAMOS-Methode verschiedene Ansätze, um sowohl die Heterogenität der Studierenden als auch den Abstraktionsgrad der Programmiersprachen und die damit verbundenen Herausforderungen im Modul anzugehen. Mit den nachfolgenden Maßnahmen verfolgen wir das Ziel, eine positive Einstellung gegenüber

dem Fach und das individuelle Wachstum der einzelnen Studierenden zu fördern sowie sicherzustellen, dass alle Studierenden die Lernziele erreichen.

Abbildung 5: Mit EMPAMOS optimierter Ablauf des Kurses »Grundlagen der Programmiertechnik«



3.1 Eingangstests zur Bestimmung der Leistungsstärke (Spielement »Aussetzen«)

Zu Beginn des Semesters absolvieren alle Studierenden einen Eingangstest, anhand dessen ihre Programmierkenntnisse bewertet werden. Basierend auf den Testergebnissen werden die Studierenden dann in zwei Gruppen eingeteilt: Anfänger:innen und Fortgeschrittene. Diese Einteilung dient als Empfehlung, doch haben die Studierenden die Freiheit, selbst zu entscheiden, welcher Gruppe sie sich zuordnen möchten. Dieses Vorgehen berücksichtigt zwei didaktische Ansätze: (1) Studierende, deren Testergebnisse an der Grenze zwischen den Gruppen liegen, können sich für diejenige Gruppe entscheiden, die ihrer Selbsteinschätzung entspricht, und (2) Studierende mit guten Testergebnissen aber geringem Vertrauen in ihre Fähigkeiten erhalten die Möglichkeit, ihre Grundkenntnisse aufzufrischen. Im weiteren Verlauf des Semesters werden die zwei Gruppen getrennt voneinander im wöchentlichen Wechsel in Präsenz unterrichtet: Während Gruppe A in der zweiten Semesterwoche Präsenzunterricht erhält, bearbeitet Gruppe B in dieser Zeit

selbstständig über das Lernmanagement-System Moodle digitale Lernpakete, welche aus Videos, Quizzes und Übungsaufgaben bestehen. In der dritten Semesterwoche wird dann Gruppe B von der Professorin unterrichtet, während sich Gruppe A den digitalen Selbstlerninhalten widmet.

Die Lerninhalte der fortgeschrittenen Gruppe sind so aufbereitet und aufeinander abgestimmt, dass die Studierenden dieser Gruppe die Lernziele des Kurses bereits zur Hälfte des Semesters erreicht haben sollten. Dieses Konzept ermöglicht es den fortgeschrittenen Studierenden, schneller voranzukommen und sich gezielt auf anspruchsvollere Themen zu konzentrieren. Gleichzeitig entstehen dadurch aber auch mehr Zeit und Raum, um die Anfänger:innen intensiver zu betreuen, was den unterschiedlichen Leistungsniveaus der Studierenden gerecht wird. Nach sechs Wochen haben die Fortgeschrittenen die Möglichkeit, das Modul durch eine Prüfung frühzeitig abzuschließen.

Nach den ersten sechs Wochen, in denen sich Präsenzlehre und digitale Selbstlernphasen abgewechselt haben, nehmen die Studierenden der Anfänger:innengruppe dann – zusätzlich zum Selbststudium mit digitalen Lernpaketen und Übungen – wieder wöchentlich an Präsenzterminen teil. Dies bietet ihnen die Möglichkeit, intensiver betreut zu werden, da die fortgeschrittene Gruppe zu diesem Zeitpunkt bereits das Modul abgeschlossen hat. Die Professorin kann sich dadurch verstärkt auf die Bedürfnisse der weniger erfahrenen Studierenden konzentrieren, was eine gezieltere Förderung ermöglicht.

3.2 Digitale Übungs- und Lehrmaterialien (Spielement »*Informationsasymmetrie*«)

Alle Lerninhalte des Moduls werden über Moodle in Form von digitalen Lernpaketen, Videos und Übungsaufgaben mit unterschiedlichen Schwierigkeitsgraden abgebildet. Somit können alle Studierenden des Kurses die Lerninhalte in ihrer eigenen Geschwindigkeit bearbeiten und bei Verständnisproblemen wiederholen. Für die Übungsaufgaben im Programmieren wird das Plugin »CodeRunner« verwendet. Dieses Plugin ermöglicht es, Codes in verschiedenen Programmiersprachen eigenständig auf Korrektheit und Funktionalität zu überprüfen und automatisch generiertes Feedback einzuholen, das auf Fehler hinweist und Lösungshinweise liefert. Dadurch können die Studierenden selbstständig und ohne Unterstützung der Lehrenden Feedback zu ihrem Lernstand erhalten. Dies ermöglicht es, individuell auf die Bedürfnisse und das Leistungsniveau der Studierenden einzugehen und somit eine ebenso effiziente wie effektive Lernumgebung zu schaffen.

Im Lehrprojekt wurden darüber hinaus diverse Maßnahmen zur Binnendifferenzierung ergriffen, um den unterschiedlichen Vorkenntnissen der Studierenden gerecht zu werden. Dazu gehörten die Einteilung der Studierenden in Anfänger:innen- und Fortgeschrittenengruppen basierend auf Eingangstests, die Bereitstellung von digitalen Lernpaketen mit unterschiedlichen Schwierigkeitsstufen, die Nutzung des CodeRunner-Plugins für individualisiertes Feedback sowie ein Level-Up-XP-System, das den Fortschritt der Studierenden visualisiert und die Lernenden dadurch zusätzlich motiviert. Diese Maßnahmen haben es ermöglicht, sowohl fortgeschrittene Studierende zu fördern als auch weniger erfahrenen Studierenden gezielte Unterstützung anzubieten.

3.3 Feedback- und Level-Up-XP-System (Spielemente »Feedback« und »Fortschrittsanzeige«)

Ein zentrales Element des Lehrkonzepts ist das Feedbacksystem über das Moodle-Plugin »Level-Up-XP«. Dieses Plugin visualisiert den Lernfortschritt der Studierenden und motiviert durch sichtbare Erfolge in Form von Erfahrungspunkten und Levelfortschritten in Relation zum Gesamtziel. An den Levelfortschritt ist eine Bonusleistung (vgl. Abschnitt 3.4) gekoppelt, die aber erst erbracht werden kann, wenn das letzte Level erreicht wurde. Regelmäßige Tests und Rückmeldungen sorgen dafür, dass Studierende kontinuierlich motiviert bleiben und ihre Leistungen verbessern können. Dies zielt besonders darauf ab, bei Studierenden mit geringen Vorkenntnissen durch kleine Erfolge mehr Selbstvertrauen aufzubauen.

Ein anonymes Leaderboard im Moodle-Kurs soll zusätzlichen Wettbewerbscharakter schaffen und die Studierenden motivieren. Dieses Leaderboard wird jedoch nur innerhalb der beiden Leistungsgruppen geführt, um einen fairen Wettbewerb zu gewährleisten. So wird verhindert, dass erfahrene Studierende die Anfänger:innen dominieren und wegen zu großer Leistungsunterschiede demotivieren.

3.4 Bonusleistungen (Spielement »Belohnungen«)

Eine Bonusleistung ist eine zusätzliche Aufgabe, die in das Modul eingebunden wird, um das Lernen und die Vertiefung des Stoffes zu fördern. Diese Bonusleistung kann genutzt werden, um das Verständnis und die Fähigkeiten der Studierenden zu vertiefen und ihre Motivation zu steigern. Bei erfolgreicher

Erbringung dieser einmal pro Person möglichen Leistung kann die Note um maximal 0,3 Punkte verbessert werden. Diese Bonusleistungen sind freiwillig und dienen nicht als Ersatz für die reguläre Modulprüfung. Sie können jedoch genutzt werden, um das Verständnis und die Fähigkeiten der Studierenden zu vertiefen und deren Motivation zu steigern.

Die Bonusleistung für das Pflichtmodul »Grundlagen der Programmier-technik« besteht im Erstellen einer Übungsaufgabe zu einem von den Lernenden subjektiv als schwierig empfundenen Themengebiet. Diese Übungsaufgabe wird in digitaler Form über die Lernplattform Moodle bereitgestellt, sodass sie von Kommiliton:innen aus dem aktuellen Semester oder von Studierenden in den folgenden Semestern als Lern Gelegenheit genutzt werden kann. Bevor die Übungsaufgabe veröffentlicht wird, erfolgt eine Qualitätskontrolle durch die Lehrenden, um sicherzustellen, dass die Aufgabe didaktisch sinnvoll ist und den Anforderungen des Moduls entspricht.

4. Evaluation

Um die Wirksamkeit der Maßnahmen zu evaluieren, kommen sowohl qualitative als auch quantitative Methoden zum Einsatz. Zum einen wird der Eingangstest zur Feststellung des Leistungsniveaus am Ende des Semesters wiederholt. Durch den Vergleich der Ergebnisse können die Leistungszuwächse individuell aufgezeigt werden. Zusätzlich wird sowohl zu Beginn als auch am Ende des Semesters eine qualitative Befragung durchgeführt. Hierbei stützen wir uns auf einen bereits erprobten Fragebogen (Amnouchokanant et al., 2021), der die Einstellungen von Erstsemesterstudierenden zum Programmieren ermittelt. Untersucht werden dabei verschiedene Dimensionen wie Interesse, Selbstwirksamkeit, Kreativität, Zusammenarbeit und Sinnhaftigkeit. Die vor und nach der Maßnahme erhobenen Werte werden miteinander verglichen, um Veränderungen und die Wirkung der eingesetzten Lehrmethoden zu analysieren. Die erste Evaluationsrunde wird zum Ende des aktuellen Semesters stattfinden, sobald alle Studierenden die abschließenden Prüfungen abgelegt haben. Dadurch können die Ergebnisse der Eingangstests und Befragungen direkt mit den Abschlusstests und dem Feedback der Studierenden verglichen werden.

Diese Mixed-Methods-Vorgehensweise, die sowohl qualitative als auch quantitative Daten berücksichtigt, stellt sicher, dass ein umfassendes Bild der Lernfortschritte und Einstellungen seitens der Studierenden gewon-

nen werden kann. Die Kombination von quantitativen und qualitativen Daten ermöglicht es uns, sowohl messbare Leistungsverbesserungen als auch subjektive Veränderungen in der Wahrnehmung und Motivation der Studierenden zu erfassen. So können wir gezielt auf die Bedürfnisse der Studierenden eingehen und die Lehrmethoden entsprechend anpassen, um eine nachhaltige Verbesserung des Lernerfolgs zu erzielen.

Abbildung 6: Transferierbare Ergebnisse



Literatur

- Amnouychokanant, V., Boonlue, S., Chuathong, S. & Thamwipat, K. (2021). A study of first-year students' attitudes toward programming in the innovation in educational technology course. *Education Research International*, 2021, e9105342. <https://doi.org/10.1155/2021/9105342>
- Feldhaus, M. & Speck, K. (2020). Die Bedeutsamkeit unterschiedlicher Lebensbereiche für den Studienerfolg. In M. Feldhaus & K. Speck (Hg.), *Herkunftsfamilie, Partnerschaft und Studienerfolg* (S. 9–36). Ergon-Verlag.
- Figueiredo, J. & García-Peñalvo, F. (2021). Teaching and learning tools for introductory programming in university courses. In A. Balderas, A. J. Mendes & J. M. Dodero (Hg.), *International Symposium on Computers in Education (SIIE)* (S. 1–6). IEEE.

Voit, T., Zinger, B. & Bröker, T. (2022). Spielfeld Lehre: die Lehre anders denken lernen. In Claudia W. & P. Riegler (Hg.), *Forum der Lehre: Vielfalt leben – Heterogenität in Studium und Lehre. Tagungsband zum Forum der Lehre an der OTH Regensburg, 4. Mai 2022* (S. 116–123). BayZiel.



Bildquelle: »Artificial Illustrations« – ein studentisches Projekt des FIDL

