ISSN: 2190-4790

LBzM Ludwigsburger Beiträge zur Medienpädagogik

Online-Magazin des Interdisziplinären Zentrums für Medienpädagogik und Medienforschung an der PH Ludwigsburg

Zwischen Wollen, Wissen und Können: Ein modularer KI-Mentor als adaptive Stabilisierungsinstanz in projektbasierten Lernprozessen der Hochschullehre

Rouven Süpfle

Zusammenfassung

Der Beitrag dokumentiert die Entwicklung und erste Evaluation eines modularen KI-Mentors für ein interdisziplinäres Hochschulseminar. Der KI-Mentor wurde vom Autor eigens für diesen Seminarkontext konzipiert und umgesetzt. Er begleitet Studierende bei der Entwicklung eigener Lernspiele zu Themen der Demokratiebildung. Dafür umfasst er vier spezialisierte KI-Module für Planung, Didaktik, Game Design und Coding. Alle Module greifen auf dasselbe Sprachmodell zu. Je Modul steuern zwei aufgaben- und modusspezifische Systemprompts den Wissensaufbau und die Wissensabsicherung.

Ausgangspunkt ist der Befund aus vorherigen Seminarzyklen, dass Studierende Schwierigkeiten haben, erreichbare Erwartungen an Programmierung, Game Design und die Integration fachdidaktisch fundierter Lernziele miteinander in Einklang zu bringen. Diese Schwierigkeiten sind vor dem Hintergrund ihrer Ausbildungsphase und der Komplexität der Aufgabe zu verstehen (in der professionellen Praxis werden solche Herausforderungen schließlich von spezialisierten Teams arbeitsteilig bewältigt). Dieses Spannungsverhältnis wird theoretisch mit zwei Modellen gefasst: dem PECC-Modell, das Lernprozesse beim Programmieren in die Phasen Playing, Engagement, Creativity und Creating gliedert, sowie dem Verhältnis von Wollen, Wissen und Können, dessen drei Dimensionen sich wechselseitig bedingen.

Die Datenerhebung umfasste vier Messzeitpunkte während des Seminars und eine Abschlussbefragung. Beteiligt waren Studierende aus vier europäischen Hochschulen. Die Antworten deuten darauf hin, dass der KI-Mentor bei Planungsaufgaben Orientierung bieten kann. Grenzen zeigen sich bei der Flexibilität des Systems, der Kontextübernahme zwischen den Modulen und der Unterstützung beim Programmieren.

Schlüsselbegriffe: ● *Modulares KI-Mentoring-System* ● *Serious Games* ● *Design-Based Research* ● *Computational Thinking* ● *Hochschuldidaktik* ● *Demokratiebildung* ● *Projektbasiertes Lernen*

1. Einleitung

Wie lassen sich kreative Freiräume und didaktische Strukturierung in der Hochschullehre so verbinden, dass sie sich gegenseitig stärken? Diese Frage bildet den Ausgangspunkt des vorliegenden Beitrags. Er untersucht ein interdisziplinäres Seminar an der Pädagogischen Hochschule Ludwigsburg (PH Ludwigsburg), in dem Lehramtsstudierende eigene Serious Games zu Themen der Demokratiebildung gestalten (vgl. Hauk/May 2025, S. 237). Wie sie bei dieser Form der eigenständigen Spielentwicklung unterstützt werden können, ist bislang wenig untersucht. Der entwickelte Ansatz knüpft an Überlegungen zum spielbasierten Lernen an, für das James Paul Gee (2005, S. 5 f.) zentrale Lernprinzipien digitaler Spiele beschreibt.

Die Entwicklung eigener Lernspiele ist eine komplexe Lernaufgabe, bei der fachliche, didaktische und technische Entscheidungen aufeinander abgestimmt werden müssen. Eine ähnliche Abstimmung ist bei der Unterrichtsplanung erforderlich, wenn Lehramtsstudierende Lernziele, Inhalte und Methoden aufeinander beziehen. Die dafür notwendigen fachlichen Kenntnisse und praktischen Erfahrungen bauen sie im Verlauf ihrer Ausbildung auf und erweitern sie. Lehramtsstudierende setzen sich dabei häufig Ziele, die ihre bisherigen Umsetzungsmöglichkeiten übersteigen. Im Projektverlauf müssen sie diese Ziele schrittweise anpassen, was zu Enttäuschungen führen kann. Erste Interviewdaten aus drei Seminarzyklen bestätigen dieses Muster (vgl. Süpfle 2025). Der vorliegende Beitrag dokumentiert einen weiteren DBR-Zyklus (Design-Based Research) einer fortlaufenden Studie. In diesem Zyklus wurde ein modularer KI-Mentor als neue didaktische Intervention entwickelt, implementiert und evaluiert. Technische Fortschritte im Bereich generativer KI erleichterten im Projektverlauf die Umsetzung des KI-Mentors.

Im ersten Versuch sollten die Studierenden eigens erstellte Unterrichtsentwürfe in Lernspiele übertragen. Dahinter stand die Annahme, Unterrichtsphasen wie Motivation, Erarbeitung und Sicherung könnten ein tragfähiges Grundgerüst für den Verlauf eines Lernspiels bilden. Da die entstandenen Spiele meist einen eigenen Plot entwickelten und deutlich von den Unterrichtsentwürfen abwichen, wurde diese Vorgabe aber aufgegeben. Im zweiten Durchlauf erhielt Game Design mehr Raum. Die Studierenden konnten den Scratch-Prototyp „Frog And His Swamp“ als Strukturvorlage nutzen, verfolgten jedoch meist aus eigenem Antrieb eigene Spielideen. Deren Umsetzung war für viele mit hohem Arbeitsaufwand verbunden. Im folgenden Durchlauf wurde der vorstrukturierte Prototyp deshalb zur verbindlichen Grundlage der Spielentwicklung, hinzu kam ein Praxistest an einer Schule. Der Prototyp setzt eine Fallanalyse nach Sibylle Reinhardt (2022) um. Die Studierenden sollten den bereitgestellten Code zunächst analysieren, seine Struktur auf CRC-Cards übertragen und ihn anschließend nachprogrammieren (vgl. Geldreich/Talbot/Hubwieser 2019, S. 184).

Scratch ist eine blockbasierte Programmierumgebung, in der Programme aus visuell kombinierbaren Blöcken zusammengesetzt werden. Dadurch ist sie für Einsteigende leichter zugänglich als textbasierte Programmiersprachen. Die Arbeit mit visuellen Codeblöcken birgt jedoch das Risiko einer trügerischen Einfachheit. Kashif Amanullah und Tim Bell (2019) zeigen, dass die Mehrheit der Scratch-Nutzenden selbst über längere Zeiträume keine fortgeschrittenen Programmierstrategien einsetzt. Dies legt nahe, auch in blockbasierten Umgebungen Computational Thinking, also die strukturierte Problemlösung auf Grundlage informatischer Konzepte (vgl. Wing 2006), gezielt zu fördern (vgl. Brennan/Resnick 2012).

Auch beim Einsatz generativer KI bleibt informatisches Vorwissen bedeutsam: In einer Studie mit Studierenden gingen höhere Werte in einem Informatiktest mit besseren Ergebnissen beim KI-gestützten Programmieren einher (vgl. Thorgeirsson/Weidmann/Su 2026). Ergänzend erhielten die Studierenden deshalb Begleitmaterialien wie Erklärvideos und CRC-Cards (Class-Responsibility-Collaboration). Diese tabellarischen Planungskarten beruhen auf einer für Scratch angepassten Methode der objektorientierten Softwareentwicklung (vgl. Keller et al. 2019). Mit ihnen lassen sich Figuren (Sprites), ihre Aufgaben und Interaktionen sowie benötigte Variablen vor dem Programmieren planen. Die inhaltliche Ausgestaltung erforderte weiterhin eigenständige Denkleistungen, während das Programmieren weitgehend reproduktiv erfolgte. Die Rückmeldungen der Studierenden deuteten darauf hin, dass die stärkere Strukturierung die fachdidaktische Umsetzung unterstützte und von vielen als entlastend erlebt wurde. Gleichzeitig zeigte sich, dass durch die stärkere Strukturierung zentrale Potenziale verloren gingen, insbesondere für kreative Offenheit und eigenständige Gestaltung. Daher stellt sich die leitende Frage des Beitrags: Wie lässt sich die Projektarbeit didaktisch strukturieren, ohne kreative Prozesse zu beschneiden?

Zunächst wurde den Studierenden ein einzelner Custom GPT über einen öffentlichen Link zugänglich gemacht. Erste Versuche damit scheiterten an Nutzungslimits und daran, dass die Bereitstellung des Links allein kaum dazu anregte, das Angebot zu nutzen. Daraufhin wurde die Problemstellung mithilfe eines Design-Thinking-Prozesses schrittweise neu gefasst. Daraus entstand ein modularer KI-Mentor, der für unterschiedliche Teilaufgaben des Entwicklungsprozesses jeweils gezielte Unterstützung bietet.

Für den Einsatz von KI in der Hochschullehre muss eine didaktische Rahmung entwickelt werden. Im beschriebenen Seminar wurde der Ansatz verfolgt, die Studierenden mittels KI dabei zu unterstützen, die komplexe Aufgabe der Lernspielentwicklung im Sinne des Computational Thinking in lösbare Teilprobleme zu zerlegen. Diese sollten strukturiert bearbeitet und die Teillösungen zu einem Produkt zusammengeführt werden. Bei der Lernspielentwicklung greifen diese Aufgaben ineinander: Aus dem Lernziel werden Etappenziele abgeleitet und in Spielsequenzen übersetzt. Dafür sind Designentscheidungen zu treffen, etwa darüber, ob Spiel- oder Dialogverläufe linear oder verzweigt gestaltet werden. Dies weist Parallelen zur Unterrichtsplanung auf. Für die Programmierung müssen solche Entscheidungen in ausführbare Abläufe und Regeln übersetzt werden. Dies kann dazu anregen, didaktische Planungsentscheidungen zu präzisieren und ihre Konsequenzen zu prüfen. Auch die anschließende Programmierung wird zunächst in Teilschritte zerlegt. Computational Thinking bezeichnet hier dieses strukturierende Vorgehen und ist nicht mit der Beherrschung einer Programmiersprache gleichzusetzen. Der modulare KI-Mentor folgt dieser Logik, indem er bereichsspezifische Unterstützung bietet, den Wissensaufbau fördert und vorhandenes Wissen festigt, ohne Lernprozesse zu ersetzen oder Ergebnisse vorwegzunehmen.

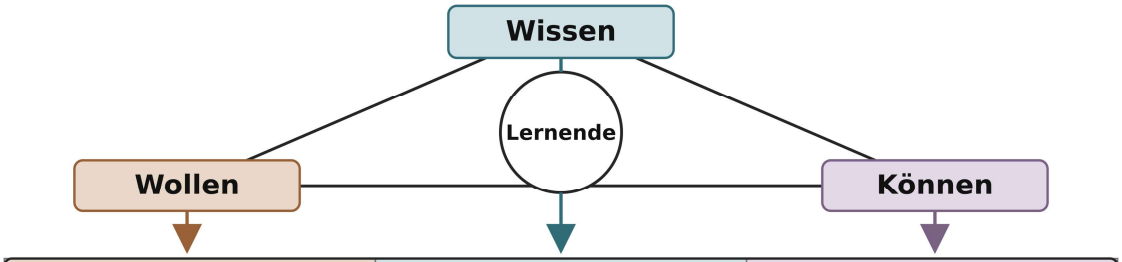
Anknüpfend an eigene Befunde (vgl. Süpfle 2025), die den Bedarf an einer stabilisierenden Intervention empirisch belegen, werden folgende Forschungsfragen untersucht: Inwiefern und durch welche Mechanismen stabilisiert ein modularer KI-Mentor den designbasierten Spielentwicklungsprozess? Wie unterscheiden sich die mittleren Selbsteinschätzungen zu Wollen, Wissen und Können zwischen den vier Messzeitpunkten? Welche Momente im Arbeitsprozess erleben die Teilnehmenden als instabil und welchen Formen der Unterstützung

schreiben sie eine stabilisierende Wirkung zu? Unter welchen Bedingungen erweist sich der KI-Mentor als wirksam und wo zeigen sich seine Grenzen? Welche Folgerungen ergeben sich daraus für die Weiterentwicklung?

2. Theoretischer Hintergrund

2.1 Das Wollen-Wissen-Können-Dreieck als Stabilitätsmodell

Die theoretische Grundlage bildet das von Hubert Sowa beschriebene Verhältnis von Wollen (bspw. Bedürfnisse, Urteile und Entscheidungen), Wissen (bspw. Daten, Fakten und Zusammenhänge) und Können (bspw. Handeln, Abläufe und Fertigkeiten). Das Wollen lässt sich demnach verwirklichen, wenn geeignetes Wissen und Können vorhanden sind (vgl. Sowa 2011, S. 18). Unklare Ziele erschweren es, das benötigte Wissen zu bestimmen. Fehlendes Können kann wiederum dazu führen, dass Ziele angepasst oder aufgegeben werden. Die in Abbildung 1 vorgenommene Konkretisierung ist exemplarisch und nicht abschließend; die aufgeführten Aspekte lassen sich ergänzen und weiter ausdifferenzieren.



Absichten und Ziele	Daten, Fakten, Zusammenhänge	Handeln, Abläufe, Fertigkeiten …
Die Bedeutsamkeit des Themas für die Zielgruppe sichtbar machen	Wissenslücken durch verlässliche Quellen schließen und fachliche Aussagen auf Triftigkeit prüfen	Fachwissen verdichten, für die Zielgruppe relevantes Orientierungs- und Detailwissen auswählen
Das Lernen der Zielgruppe fördern	Lernziele und didaktische Gestaltungsprinzipien verstehen	Passende Lernsettings zu konkret formulierten Lernzielen entwerfen
Spielspaß und fachliches Lernen verbinden	Funktionsweisen von Spielmechaniken erläutern	Spielmechaniken auswählen und passend zum Lernziel adaptieren
Funktionsfähiges Spiel vollenden	Verstehen, wie Variablen, Bedingungen und Schleifen einzeln oder kombiniert bestimmte Programm-funktionen ermöglichen	Sprites und Interaktionen mit CRC-Cards planen, Programmierkonzepte auswählen, kombinieren und in Scratch umsetzen
Die Qualität des eigenen Spiels verbessern	Kriterien zur Beurteilung von Lernspielen kennen	Lernspiele aus der Zielgruppenperspektive prüfen und gezielt überarbeiten
…	…	…

Abbildung 1: Wollen, Wissen und Können bei der Entwicklung digitaler Lernspiele (eigene Darstellung auf Grundlage von Sowa 2011, S. 18).

Im untersuchten Seminar formulieren Studierende ohne vertiefte Vorerfahrungen in Programmierung, Game Design oder fachdidaktischer Anwendung häufig ambitionierte Ziele, die ihr vorhandenes Wissen und Können übersteigen. Solche Spannungen wurden bereits im vorangehenden Beitrag beschrieben (vgl. Süpfle 2025, S. 6637 f.). Die zentrale didaktische Herausforderung besteht daher darin, den Weg von der Zielvorstellung über das benötigte Wissen bis zur Umsetzung zu unterstützen, ohne den Studierenden die notwendigen Entscheidungen abzunehmen.

Für die vorliegende Untersuchung dient das Verhältnis von Wollen, Wissen und Können als heuristischer Rahmen. Untersucht werden die Selbsteinschätzungen der Lernenden in diesen drei Dimensionen beim Einsatz des KI-Mentors. Stabilisierung bezeichnet dabei eine aus Sicht der Studierenden verbesserte Passung zwischen Wollen, Wissen und Können. Hinweise darauf liefern das Stabilisierungsitem und die offenen Angaben zu Klärungsmomenten und Handlungsfähigkeit. Die Mittelwerte der drei Kerndimensionen dienen ergänzend dem Vergleich der Selbsteinschätzungen zwischen den Messzeitpunkten.

2.2 Das PECC-Modell und seine Erweiterung

Die Konzeption des Seminars orientiert sich am PECC-Modell nach Bernadette Spieler (2022), das Lernprozesse im Bereich Computational Thinking entlang der Dimensionen Playing, Engagement, Creativity und Creating strukturiert. Das Modell verbindet gendersensible Zugänge zur Informatikbildung mit kooperativem und explorativem Lernen. Für den Seminarkontext an der PH Ludwigsburg erweist sich dieser Ansatz als grundsätzlich anschlussfähig, insbesondere vor dem Hintergrund einer heterogenen und mehrheitlich weiblichen Studierendenschaft.

Die Verbindung des PECC-Modells mit einer kunstdidaktischen Perspektive im Anschluss an Sowa wurde bereits im vorangehenden Beitrag vorgeschlagen (vgl. Süpfle 2025, S. 6637 f.). Die vorliegende Untersuchung baut darauf auf. Die kunstdidaktische Perspektive ist hier anschlussfähig, weil auch bei der Lernspielentwicklung eigene Gestaltungsideen mit den jeweiligen Umsetzungsmöglichkeiten abgestimmt werden müssen und Anlässe zum Erwerb zusätzlichen Wissens und Könnens bieten. Die ergänzende Betrachtung von Wollen, Wissen und Können richtet sich auf die Passung zwischen den Zielen der Studierenden und ihren wahrgenommenen Umsetzungsmöglichkeiten. Im Zentrum des vorliegenden Beitrags stehen die Entwicklung und Evaluation eines modularen KI-Mentors als didaktischer Intervention in einem weiteren DBR-Zyklus.

Im Projektverlauf wurde das entwickelte Seminarkonzept in einem Blended Intensive Programme (BIP) erprobt. Das Kurzprogramm verband eine virtuelle Phase mit einer gemeinsamen Präsenzphase und sollte den Studierenden internationale Perspektiven auf die Querschnittsthemen „Demokratiebildung“ und „informatische Grundbildung“ eröffnen.[1]

Die Studierenden unterschieden sich in ihrer Arbeitsweise und ihrem Arbeitstempo. Daher lassen sich die PECC-Phasen nicht für alle Studierenden einheitlich den vier Messzeitpunkten zuordnen. Elemente der Playing-Phase wurden bereits vor dem evaluierten Projektzeitraum umgesetzt, etwa in der vorgelagerten Onlinephase und bei einer gamifizierten Stadtrallye.

[1] Das Blended Intensive Programme wurde im Rahmen von Erasmus+ gefördert.

Auch während des evaluierten Projektzeitraums fanden spielerische Aktivitäten statt. Im Vordergrund standen jedoch Engagement, Creativity und Creating. Teile dieser Prozesse fanden außerhalb der Interaktion mit dem KI-Mentor statt, insbesondere in Form informeller Unterstützung durch Mitstudierende. Dies zeigt sich auch darin, dass die Peer-Group häufig als entscheidende Unterstützung genannt wurde (vgl. Kapitel 5.4). Aufgrund des kompakten Zeitrahmens konnten kooperative Arbeitsphasen, etwa durch agile Methoden, nur ansatzweise systematisch gestaltet werden.

2.3 Design-Based Research als Forschungsrahmen

Design-Based Research (DBR) verbindet die Entwicklung von Bildungsinterventionen mit deren empirischer Untersuchung in authentischen Lernkontexten. Charakteristisch ist der iterative Aufbau, bei dem Erkenntnisse aus einzelnen DBR-Zyklen in die Weiterentwicklung des Designs einfließen. Ziel ist es, Bildungsinterventionen zu verbessern und theoretische Erkenntnisse über ihre Gestaltung und ihren Einsatz zu gewinnen (vgl. Reinmann/Herzberg/Brase 2024, S. 16 f.).

Die vorliegende Untersuchung bildet den vorläufigen Abschluss einer fortlaufenden DBR-Studie, in der das Seminardesign über mehrere Seminarzyklen hinweg entwickelt und empirisch untersucht wurde. Das Forschungsdesign wurde dabei – einem iterativen Aufbau entsprechend – weiterhin an Beobachtungen aus der Durchführung angepasst. So wurde in der vorliegenden Erhebung der Fragebogen erweitert, als sich zeigte, dass Teilnehmende neben dem KI-Mentor weitere KI-Tools einsetzten. Aufgrund des kompakten Zeitrahmens lagen die vier Messzeitpunkte eng beieinander. Dadurch ließ sich erfassen, wie sich die Einschätzungen der Gruppe im Verlauf der Projektarbeit verschoben.

2.4 Phasenmodell: Vorbereitung – Planung – Durchführung – Evaluation

Für die Analyse des Seminarverlaufs werden in Anlehnung an Kerstin Klein (2008) vier Phasen unterschieden: Vorbereitung, Planung, Durchführung und Evaluation. Die Vorbereitungsphase umfasst Maßnahmen vor der eigentlichen Projektarbeit, etwa die Einführung in Scratch. Im untersuchten Seminar fand sie während der Onlinephase statt und ist daher nicht Gegenstand der Untersuchung. In der Planungsphase werden Ziele und Designentscheidungen konkretisiert, woraufhin die technische Umsetzung während der Durchführungsphase erfolgt. In der Evaluationsphase wird das Projektergebnis reflektiert.

Das Phasenmodell dient dazu, die Ergebnisse entlang des Projektverlaufs einzuordnen. Die vier Messzeitpunkte entsprechen hierbei nicht unmittelbar den einzelnen Phasen. Die Einschätzungen der Gruppe geben jedoch Hinweise auf Phasen mit stärkeren Spannungen zwischen Wollen, Wissen und Können.

2.5 KI-Mentoring in der Hochschullehre

Die Idee, Lernende durch KI-gestützte Systeme individuell zu begleiten, schließt an grundlegende Befunde zur Wirksamkeit personalisierter Unterstützung an. Benjamin Bloom (1984, S. 4 f.) beschreibt mit dem sogenannten Zwei-Sigma-Problem das Potenzial individuell ausge-

richteter Lernbegleitung gegenüber konventionellem Gruppenunterricht. Auch für intelligente Tutorsysteme berichten James Kulik und John Fletcher (2016, S. 42) positive Effekte gegenüber konventionellem Unterricht. Diese Befunde lassen sich zwar nicht unmittelbar auf große Sprachmodelle übertragen. Deren Verbreitung eröffnet jedoch neue Möglichkeiten, individualisierte Lernbegleitung in Lernarrangements einzubinden.

3. Systembeschreibung: Der modulare KI-Mentor

3.1 Entwicklungslogik und Grundprinzip

Der modulare KI-Mentor entstand aus wiederkehrenden Beobachtungen im Seminar. Die Studierenden erprobten, wie sie eigene didaktische Entwürfe und Spielideen in ein Lernspiel umsetzen konnten. Die dafür erforderlichen Kenntnisse und praktischen Erfahrungen aufzubauen und miteinander zu verbinden, war selbst Teil der Lernaufgabe. Auftretende Schwierigkeiten zeigten, an welchen Stellen sie Unterstützung benötigten. Daraus ergaben sich vier Unterstützungsbereiche: die didaktische Fundierung des Spielinhalts, die Gestaltung der Spielerfahrung im Game Design, die technische Umsetzung durch Coding und die Projektplanung. Die Projektplanung soll dabei helfen, den Arbeitsprozess im Sinne des Computational Thinking in überschaubare Teilaufgaben und Phasen zu gliedern, den erforderlichen Arbeitsaufwand innerhalb der verfügbaren Zeit zu verteilen und den Projektumfang realistisch festzulegen. Für jeden Bereich wurde ein eigenes Modul eingerichtet. Technisch greifen alle Module auf dasselbe Sprachmodell zu. Für jedes Modul wurden zwei feste Systemprompts angelegt, einer für den Wissensaufbau im Pull-Modus und einer für die Wissensabsicherung im Push-Modus. Als Basismodell kam durchgängig Claude Sonnet 4.6 über die Programmierschnittstelle von Anthropic zum Einsatz. Der Wechsel zwischen den Modulen bedeutet daher keinen Wechsel des zugrunde liegenden Modells.

3.2 Architektur: Dispatcher und spezialisierte Module

Die Systemarchitektur wurde mit der Automatisierungsplattform n8n umgesetzt. Jede Anfrage wird zunächst von einem Dispatcher klassifiziert, der dasselbe Basismodell wie die Fachmodule mit einem eigenen Systemprompt nutzt. Er ordnet einen thematischen Tag (Planning, Didactics, Design oder Coding) und einen Interaktionsmodus (Pull oder Push) zu. Die Klassifikation erfolgt modellgestützt, die anschließende Weiterleitung anhand dieser Parameter regelbasiert in n8n. Der Dispatcher arbeitet damit wie eine Vermittlungsstelle, die Anliegen an die zuständige Fachabteilung weiterleitet. Die Funktionen der beiden Interaktionsmodi erläutert der folgende Abschnitt.

3.3 Das Pull/Push-Prinzip

Das zentrale Steuerungsprinzip des Systems ist die Unterscheidung zwischen Pull- und Push-Interaktionen. Im Pull-Modus suchen Studierende Unterstützung beim Aufbau neuen Wissens. Sie formulieren Fragen und erhalten strukturierende Hinweise und leitende Rückfragen, die sie beim eigenständigen Lösen des Problems unterstützen. Im Push-Modus legen sie dagegen ihr vorhandenes Wissen dar und begründen ihre Entscheidungen. Das Modul prüft die

Erklärung, macht auf mögliche Verständnislücken aufmerksam und fordert dazu auf, das Wissen auf andere Zusammenhänge zu übertragen. Fragen Studierende beispielsweise „Wie strukturiere ich mein Spiel?“, reagiert der KI-Mentor im Pull-Modus mit leitenden Rückfragen. Im Push-Modus erläutern die Studierenden ihr Konzept und der KI-Mentor prüft, ob sie ihre Designentscheidungen begründen kann. Die Unterscheidung soll eine rein konsumierende KI-Nutzung begrenzen und die aktive Auseinandersetzung fördern. Die Zuordnung der Interaktionen wird im Kompetenzradar dargestellt.

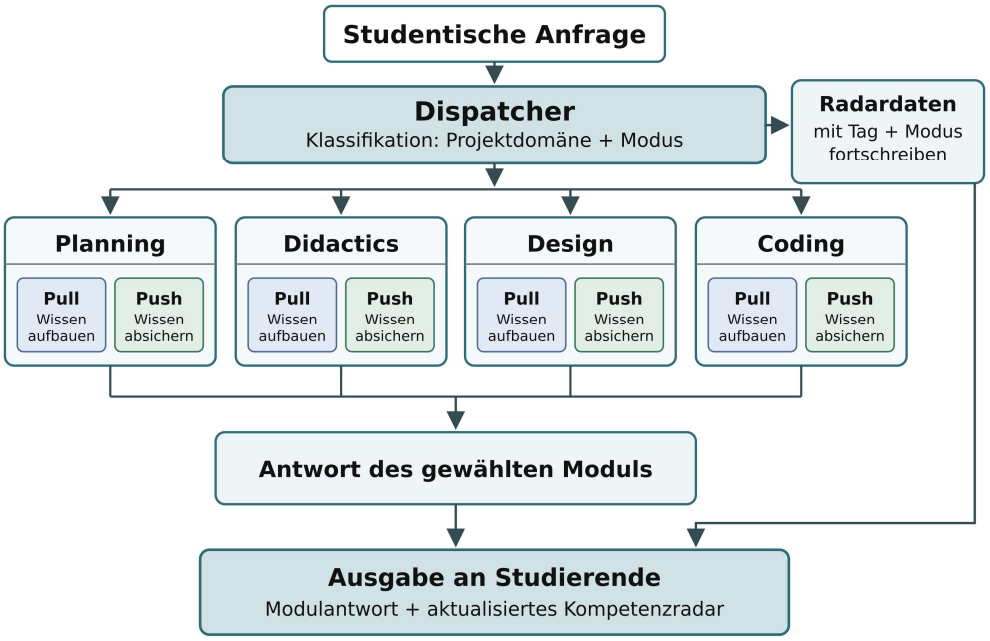


Abbildung 2: Architektur des modularen KI-Mentors.

3.4 Der Kompetenzradar

Der Kompetenzradar stellt nach jeder Interaktion deren Zuordnung zu Projektdomäne und Interaktionsmodus dar. Er zeigt, in welchen Bereichen Studierende Unterstützung suchen oder ihr Wissen darlegen, misst jedoch nicht ihre Selbstständigkeit. Als Reflexionsinstrument soll er Ungleichgewichte zwischen den Projektdomänen sichtbar machen und Auswertungsgespräche unterstützen. In anderen Lehrveranstaltungen könnte er auch als Anlass dienen, im Bewertungsgespräch Arbeitsschritte und erprobte Lösungswege gemeinsam nachzuvollziehen. Dabei wären seine Zuordnungen anhand der Erläuterungen der Studierenden und des entwickelten Spiels kritisch zu prüfen. Der Radar selbst gibt keine Leistungsbewertung vor.

Über die Schaltfläche „Bild“ konnten Studierende Screenshots mit einer Textanfrage hochladen. Das Coding-Modul konnte so beispielsweise Scratch-Blöcke analysieren und beim Debugging unterstützen. Abbildung 3 veranschaulicht die Benutzeroberfläche des KI-Mentors.

3.5 Technische Limitation: Das Gedächtnisproblem

Eine technische Grenze liegt in der fehlenden Übertragung des Gesprächskontexts zwischen den Modulen. Jedes Modul speichert seinen eigenen Gesprächsverlauf. Beim Wechsel etwa von Planning zu Coding kann das neu aktivierte Modul nicht auf den bisherigen Austausch

zugreifen. Für Studierende kann dadurch der Eindruck entstehen, der KI-Mentor sei inkonsistent oder habe Angaben vergessen (vgl. Kapitel 6.3). Ein gemeinsamer Kontextspeicher ist deshalb eine zentrale technische Weiterentwicklung.

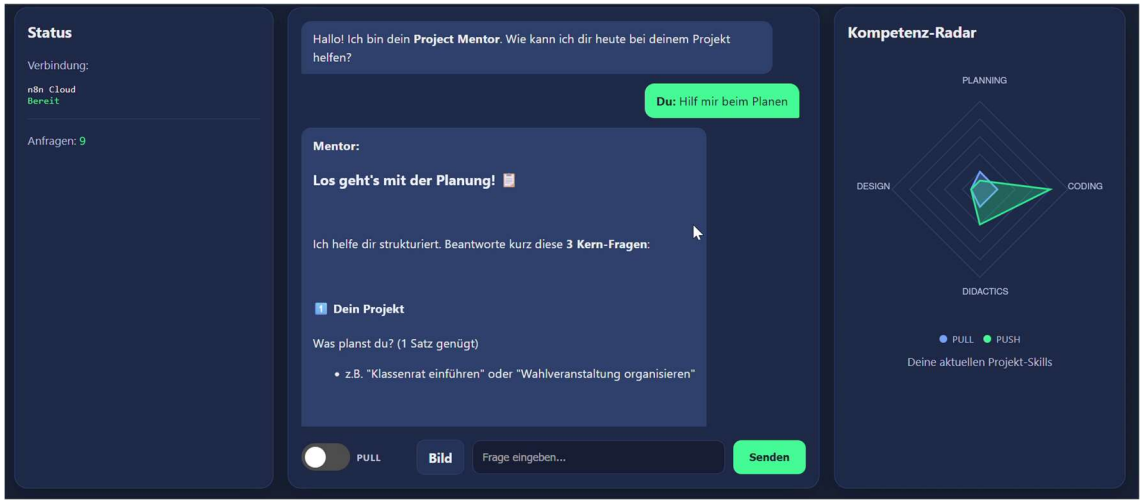


Abbildung 3: Demonstrationsansicht der Benutzeroberfläche des KI-Mentors mit Planungsdialog und Kompetenzradar.

4. Methode

4.1 Studiendesign und Erhebungskontext

Die Studie wurde im Rahmen des BIP an der PH Ludwigsburg durchgeführt. An der untersuchten Veranstaltung nahmen Studierende der PH Ludwigsburg (Deutschland), der Josip Juraj Strossmayer University of Osijek (Kroatien), der Mondragon University (Spanien) und des University College Copenhagen (Dänemark) teil.

In der vorgelagerten Onlinephase erhielten die Studierenden eine Einführung in Scratch und setzten erste kleine Projekte unter Einbezug von CRC-Cards um. Zu Beginn der Arbeit mit dem KI-Mentor erkundeten sie gemeinsam mit der Seminarleitung vor allem das Planning-Modul. Dabei wurde ihnen ein fragenbasierter Zugang gezeigt, über den sie die Arbeitsweise und Unterstützungsmöglichkeiten des Mentors eigenständig erkunden sollten.

Die Projektphase umfasste zwei Präsenztage mit jeweils zwei Arbeitsphasen. Daraus ergaben sich vier Messzeitpunkte am Vor- und Nachmittag beider Tage. Die Datenerhebung fand jeweils unmittelbar nach einer Arbeitsphase statt, damit die Erfahrungen aus dieser Phase noch präsent waren. Begleitende Beobachtungen des Seminarverlaufs wurden zur Kontextualisierung der Fragebogendaten herangezogen, jedoch nicht als eigenständiger Datensatz systematisch ausgewertet.

4.2 Stichprobe

Die Teilnehmenden wurden eingeladen, zu jedem der vier Messzeitpunkte einen Fragebogen auszufüllen. Über alle vier Messzeitpunkte hinweg wurden insgesamt 94 Fragebogenrückläufe erfasst. Nach der Bereinigung des Datensatzes verblieben 85 Antworten für die quantitative Likert-Analyse. Ausgeschlossen wurden Einträge mit der Angabe „No AI use today“, da sie keine Grundlage für die Bewertung der Nutzungserfahrung des KI-Mentors bieten. Diese

Ausschlussregel wurde im Verlauf der Erhebung festgelegt. Die offenen Antworten dieser Einträge flossen dennoch in die qualitative Analyse ein, da sie Hinweise auf Nutzungshindernisse und Kontextbedingungen geben.

Lediglich ein Fall vom Messzeitpunkt 1 wurde abweichend behandelt. Obwohl dort „No AI use today“ angegeben wurde, weisen die offenen Antworten und die Zuschreibung der entscheidenden Unterstützung eindeutig auf eine tatsächliche Nutzung hin. Der Fall verblieb deshalb in der quantitativen Analyse. Ein weiterer Antwortbogen vom vierten Messzeitpunkt enthielt keine gültigen Likert-Angaben und blieb für diese Analyse unberücksichtigt.

Zusätzlich wurde am Ende des BIP ein retrospektiver Gesamtfragebogen eingesetzt, den 21 Teilnehmende ausfüllten. Er erfasst unter anderem die selbsteingeschätzte KI-Kompetenz vor und nach dem Seminar sowie eine abschließende Bewertung der erfahrenen Unterstützung.

4.3 Erhebungsinstrument

Der sitzungsbezogene Fragebogen wurde für diesen Kontext entwickelt und auf Englisch eingesetzt. Er erfasst Hochschulstandort, Arbeitsfokus und sechs Items auf einer sechsstufigen Likert-Skala (1 = „Disagree“, 6 = „Agree“). Der Arbeitsfokus wurde mit „Didactics / Content“, „Coding / Debugging“ oder „Both“ erfasst. Zusätzlich konnte „No AI use today“ angegeben werden. Die Auswahl ermöglicht eine grobe Unterscheidung zwischen didaktisch-inhaltlicher Arbeit und technischer Umsetzung. Erfragt wurde der Schwerpunkt der Sitzung, nicht die Nutzung einzelner Module. Planung und Game Design wurden nicht gesondert erfasst. Die Items zu Wollen („Goals: Clarity on didactic/content objectives.“) und Wissen („Knowledge: Clarity on concrete implementation steps.“) erfassen die wahrgenommene Klarheit über didaktisch-inhaltliche Ziele beziehungsweise konkrete Umsetzungsschritte. Können wird mit „Confidence: I feel able to implement the next steps.“ operationalisiert und bezeichnet die wahrgenommene Handlungsfähigkeit. Zwei weitere Items betreffen die Erwartungskonformität der Antworten des KI-Mentors („System: AI responses matched my expectations.“) und die wahrgenommene Steuerbarkeit der Interaktion durch Prompts („Interaction: I could steer the AI effectively via prompts.“). Das ergänzende Stabilisierungsitem „Stabilization: I aligned goals, knowledge & implementation.“ erfasst, inwieweit die Studierenden ihre Ziele, ihr Wissen und die Umsetzung als aufeinander abgestimmt einschätzten. Offene Fragen erfassen Schwierigkeiten, Klärungsmomente und die Bewertung der Unterstützungsleistung des KI-Mentors. Ein Mehrfachauswahl-Item erfragt, welchen Quellen die Studierenden ihre Lernerfahrung zuschreiben, etwa dem KI-Mentor, Peers oder Dozierenden.

Die am ersten Messzeitpunkt analog erhobenen Daten wurden händisch digitalisiert. Ab Messzeitpunkt 3 wurde die Antwortoption „AI in general“ ergänzt, nachdem beobachtet wurde, dass Teilnehmende parallel generische KI-Tools nutzten. Diese Anpassung entspricht der Logik des DBR. Für die Wiedergabe der offenen Antworten wurden offensichtliche Tipp- und Grammatikfehler korrigiert. In den Tabellen stehen Z1 bis Z4 für die vier Messzeitpunkte. Auch qualitative Zitatnachweise behalten diese Kennungen: (Z1,4) bezeichnet Antwort 4 am ersten Messzeitpunkt, keine über die Messzeitpunkte hinweg identifizierbare Person.

4.4 Auswertungsverfahren

Die Antworten auf die sechsstufigen Likert-Items wurden in SPSS übertragen. Für jedes Item und jeden Messzeitpunkt wurden Mittelwerte (M) und Standardabweichungen (SD) aus den jeweils gültigen Antworten berechnet. Die Mittelwerte beschreiben die durchschnittliche Zustimmung, die Standardabweichungen die Streuung der Antworten. Zusätzlich wurde ausgezählt, welche Form der Unterstützung die Studierenden jeweils als entscheidend nannten. Der am Ende eingesetzte Gesamtfragebogen ergänzt die sitzungsbezogenen Daten um eine zusammenfassende Einschätzung des Seminarverlaufs.

Die offenen Antworten wurden in Anlehnung an die qualitative Inhaltsanalyse nach Philipp Mayring (2022) ausgewertet. Dazu wurden Kategorien aus dem Material entwickelt und die Antworten anschließend manuell zugeordnet. Mehrfachcodierungen waren möglich, wenn eine Antwort mehrere inhaltlich unterscheidbare Aspekte enthielt.

Der Autor erarbeitete die inhaltlichen Konzepte des Beitrags: Seminargestaltung, KI-Mentor-Entwicklung, Datenerhebung und -auswertung. Im Erarbeitungs-, Auswertungs- und Überarbeitungsprozess kamen ChatGPT (OpenAI) und Claude (Anthropic) unterstützend zum Einsatz. KI wurde als dialogisches Reflexionswerkzeug genutzt, um das vom Autor entwickelte Kategoriensystem und die Ergebnisinterpretation kritisch zu hinterfragen. Beide KI-Systeme unterstützten zudem die Kontrolle statistischer Kennwerte. Die einbezogenen Erhebungsdaten wurden daraufhin durchgesehen, ob sie eine Identifizierung einzelner Teilnehmender ermöglichten. Entsprechende Angaben wurden dabei nicht festgestellt.

Die KI-Systeme unterstützten außerdem die sprachliche und strukturelle Überarbeitung des Manuskripts sowie die kritische Durchsicht der Argumentation. Analoge Überarbeitungen durch den Autor bildeten jeweils die Diskussionsgrundlage für den folgenden Überarbeitungszyklus. Argumente wurden phasenweise mit beiden KI-Systemen diskutiert. Der Autor prüfte die Anregungen und Änderungsvorschläge am Datenmaterial sowie anhand des theoretischen Rahmens und des Forschungsdesigns, überarbeitete sie bei Bedarf und entschied über ihre Übernahme. Die Auswertungs- und Interpretationsentscheidungen traf der Autor; er verantwortet somit die fachlichen Aussagen und Schlussfolgerungen des Beitrags.

5. Ergebnisse

5.1 Kerndimensionen im Vergleich der Messzeitpunkte

Die Mittelwerte von Wollen, Wissen und Können verändern sich über die vier Messzeitpunkte (vgl. Tabelle 1 und Abbildung 4). Nach mittleren Ausgangswerten sinken am Messzeitpunkt 2 Wissen[2] und Können, während Wollen leicht ansteigt. Zum Messzeitpunkt 3 liegen die Mittelwerte aller drei Dimensionen höher als am Messzeitpunkt 2. Am Messzeitpunkt 4 fallen alle drei Mittelwerte höher aus als an den ersten beiden Messzeitpunkten. Können erreicht mit M = 4,67 seinen Höchstwert. Bei Wollen, Wissen und Können lagen die Standardabweichungen an allen vier Messzeitpunkten über einem Skalenpunkt (SD = 1,09 bis 1,44 auf der Skala von 1 bis 6).

[2] Die Studierenden verfügten über unterschiedliches Vorwissen, was in dieser Untersuchung aber nicht systematisch erhoben wurde.

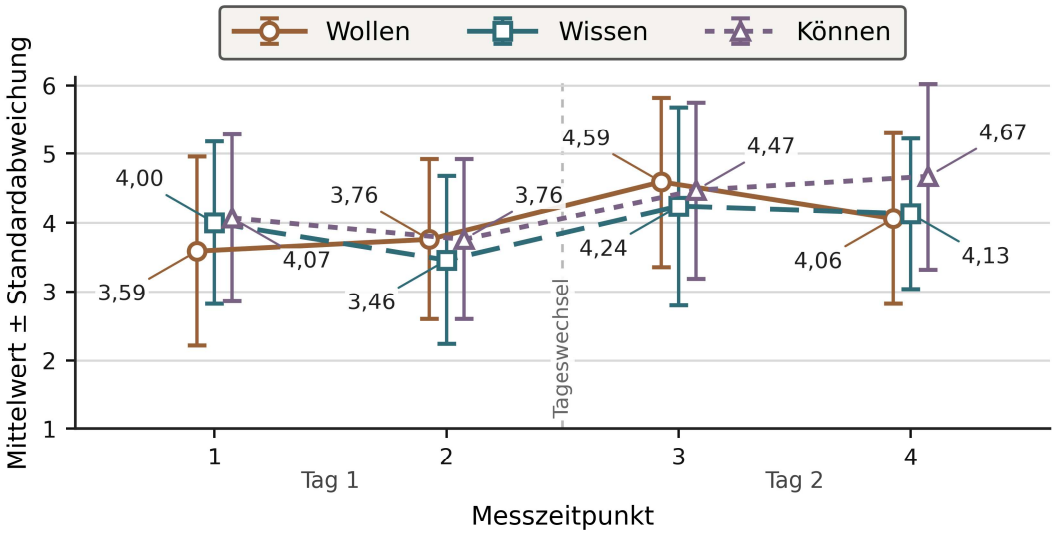


Abbildung 4: Gruppenmittelwerte von Wollen, Wissen und Können an vier Messzeitpunkten (wiederholte Querschnitte).

Beim ergänzenden Stabilisierungsitem bewerteten die Studierenden, inwieweit sie Ziele, Wissen und Umsetzung miteinander in Einklang gebracht hatten. Die mittlere Zustimmung betrug M = 4,07 am ersten, M = 3,76 am zweiten, M = 4,29 am dritten und M = 4,31 am vierten Messzeitpunkt (vgl. Tabelle 1).

Tabelle 1: Mittelwerte und Standardabweichungen der Kerndimensionen und des ergänzenden Stabilisierungsitems über vier Messzeitpunkte.

Dimension	Z1 (N=27)	Z2 (N=25)	Z3 (N=17)	Z4 (N=16)
Wollen	3,59 ±1,37	3,76 ±1,16	4,59 ±1,23	4,06 ±1,24
Wissen	4,00 ±1,18	3,46 ±1,22	4,24 ±1,44	4,13 ±1,09
Können	4,07 ±1,21	3,76 ±1,16	4,47 ±1,28	4,67 ±1,35
Stabilisierungsitem	4,07 ±1,00	3,76 ±1,09	4,29 ±1,36	4,31 ±1,25

Anmerkung: Angegeben sind M ± SD auf der Skala von 1 bis 6. N bezeichnet die Zahl bereinigter Antwortbögen. Abweichende gültige Itemantworten: Wissen Z2 n = 24, Können Z4 n = 15.

Im Phasenmodell (vgl. Kapitel 2.4) lässt sich Messzeitpunkt 2 gegen Ende der Planungsphase einordnen. Der Rückgang der Mittelwerte von Wissen und Können könnte auf realistischere Selbsteinschätzungen nach eigenständiger Projektarbeit zurückgehen, aber auch allein auf eine veränderte Zusammensetzung der Antwortenden. Der Anstieg am Messzeitpunkt 3 fällt mit dem Übergang zur Durchführung zusammen. Mit zunehmender technischer Umsetzung könnte die orientierende Funktion des KI-Mentors weniger im Vordergrund stehen.

Die Verteilung der Arbeitsfokusse stützt diese Einordnung. Berücksichtigt wurden gültige Fokusangaben ohne „No AI use today“. Am Messzeitpunkt 1 nennen 18 von 25 gültigen Angaben (72 %) Didactics / Content als Arbeitsfokus. Am Messzeitpunkt 4 entfallen zehn von

17 Angaben (59 %) auf Coding / Debugging und sieben (41 %) auf Both. Die in der Seminarplanung erwartete Verlagerung von didaktischen und inhaltlichen Fragen zur technischen Umsetzung wird damit auch in den Antworten sichtbar. Zugleich fällt der stärkste Anstieg der drei Kerndimensionen am Messzeitpunkt 3 mit dem Tageswechsel zusammen. Da sich Phasenübergang und Tageswechsel überlagern, lassen sich ihre jeweiligen Einflüsse nicht voneinander trennen. Diese Überlagerung wird in Kapitel 6.1 diskutiert.

5.2 Systembezogene Dimensionen: Erwartungskonformität und Steuerbarkeit

Die beiden systembezogenen Items erfassen die Zustimmung zu den Aussagen, dass die Antworten des KI-Mentors den Erwartungen entsprachen (Item System) und dass die Teilnehmenden die Interaktion mit dem KI-Mentor durch Prompts wirksam steuern konnten (Item Interaction; vgl. Tabelle 2). Beide Mittelwerte erreichen am ersten Messzeitpunkt ihren höchsten Wert und sinken am zweiten Messzeitpunkt. Während sich die Bewertung der Steuerbarkeit am dritten Messzeitpunkt teilweise erholt und am vierten Messzeitpunkt leicht zurückgeht, sinkt die Bewertung der Erwartungskonformität über alle vier Messzeitpunkte hinweg.

Tabelle 2: Mittelwerte und Standardabweichungen der systembezogenen Items.

Dimension	Z1 (N=27)	Z2 (N=25)	Z3 (N=17)	Z4 (N=16)
System	4,00 ±1,20	3,60 ±1,15	3,24 ±1,48	3,19 ±1,38
Interaction	4,52 ±1,09	3,44 ±1,33	3,71 ±1,40	3,63 ±1,20

Anmerkung: M ± SD (Skala 1 bis 6). N = Anzahl bereinigter Antwortbögen. Beim Item System beruhen die Werte am ersten Messzeitpunkt auf 26 gültigen Antworten.

Während Wollen, Wissen und Können am Messzeitpunkt 4 höher liegen als am ersten Messzeitpunkt, fallen die systembezogenen Bewertungen niedriger aus. Eine mögliche Erklärung ist eine differenziertere Einschätzung der Möglichkeiten und Grenzen des KI-Mentors. Die quantitativen Daten allein klären diese Entwicklung allerdings nicht. Die in offenen Antworten kritisierte fehlende Flexibilität und das unzureichende Gesprächsgedächtnis liefern weitere Erklärungsansätze.

5.3 Auswertung der offenen Antworten

Die Kategorien der offenen Antworten lassen sich drei Bereichen zuordnen: system- und lernseitige Schwierigkeiten, Klärungsmomente und Bewertungen der Unterstützungsleistung. Bei den Schwierigkeiten beschreiben Studierende eine eingeschränkte Flexibilität des KI-Mentors: „It was very (probably too) rigid in its understanding of the goals and implementation of my project“ (Z2,9). Einzelne Antworten berichten, dass aufgrund solcher Einschränkungen generische KI-Tools genutzt wurden. In der Umsetzungsphase betrafen Schwierigkeiten auch die Unterstützung beim Programmieren.

Eine Antwort vom dritten Messzeitpunkt zeigt, wie sich der Bedarf an Coding-Unterstützung und eine als starr wahrgenommene Modulgrenze überlagerten: „I wanted to start to code

in scratch, but the AI mentor did not want to help me, because it wanted me to stay in the 'designing' state" (Z3,20). Weitere systemseitige Schwierigkeiten betreffen Kontextverlust und fehlendes Gedächtnis. Aufseiten der Studierenden wurden Schwierigkeiten bei Design, Planung und Didaktik sowie bei der Orientierung beschrieben.

Ein Teil der beschriebenen Klärungsmomente betrifft Zielklärung, Projektstruktur und Orientierung. Eine Antwort beschreibt konkret die Klärung der Funktion einer zentralen Variable im Projekt: „The moment of clarity occurred when I was able to explicitly define the core variable and isolate its function with the overall system" (Z1,1). Daneben wird ein verändertes Verständnis der Rolle des KI-Mentors beschrieben. Eine Antwort ordnet ihn als Planungshilfe ein: „It changed my way of viewing the AI mentor as a planner rather than a code solver" (Z1,12). Andere Antworten berichten von keinem Klärungsmoment oder davon, den KI-Mentor nicht genutzt zu haben.

Als konstruktive Hilfe wurden Antworten codiert, in denen der KI-Mentor als orientierend, anregend oder konkret unterstützend beschrieben wurde. Die Unterstützung betraf beispielsweise Ideen, Struktur und Design: „General brainstorming is where the tutor helped me the most. I didn't have many ideas in regards to the design and it's overview. The AI tutor helped me out immensely in that aspect" (Z1,4). Begrenzte Hilfe wurde als vorhandene, aber nicht hinreichend passgenaue Unterstützung beschrieben. Der sokratische Dialogstil wurde ambivalent bewertet: „It wasn't unhelpful, we are just not used to an AI that asks that many questions in general, it makes you think a lot" (Z2,25). Aus der Nicht-Nutzung allein lässt sich aber keine negative Bewertung des KI-Mentors ableiten.

5.4 Attribution: Wem schreiben die Studierenden Unterstützung zu?

Tabelle 3 zeigt, welche Unterstützungsquellen die Studierenden jeweils als entscheidend nannten. Berücksichtigt wurden alle gültigen Angaben, auch aus Sitzungen ohne Nutzung des KI-Mentors. Die Zahlen beziehen sich auf Antworten über vier Messzeitpunkte, nicht auf entsprechend viele verschiedene Personen. Combination umfasst die direkte Wahl dieser Option sowie Mehrfachnennungen und kann den KI-Mentor einschließen. AI Mentor erfasst dessen alleinige Nennung.

Tabelle 3: Zuschreibung der entscheidenden Unterstützung über vier Messzeitpunkte.

Kategorie	Z1 (N=26)	Z2 (N=25)	Z3 (N=21)	Z4 (N=19)	Gesamt
AI Mentor	6	4	1	1	12
Peer Group	1	4	4	4	13
Instructor	0	3	2	2	7
Combination	19	14	6	8	47
AI in general*	0	0	8	4	12

*Antwortoption ab Messzeitpunkt 3 verfügbar

Insgesamt überwiegt die Zuschreibung an eine Kombination verschiedener Unterstützungsquellen (vgl. Tabelle 3). Die alleinige Nennung des KI-Mentors geht zurück, während die Peer Group ab Messzeitpunkt 2 häufiger genannt wird. An den Messzeitpunkten 3 und 4 wird auch die dann verfügbare Option AI in general gewählt. Die Verteilung verweist auf das Zusammenspiel menschlicher und KI-gestützter Unterstützung im Seminar. Sie bildet keine Nutzungshäufigkeiten ab.

5.5 Retrospektive Gesamtperspektive

Der retrospektive Abschlussfragebogen (N = 21) ergänzt die prozessbezogenen Daten. Am Seminarende schätzten die Studierenden ihre KI-Kompetenz rückblickend für die Zeit vor dem Seminar mit M = 4,00 (SD = 1,00) und für die Zeit danach mit M = 4,43 (SD = 0,87) ein. Beide Angaben wurden zum selben Zeitpunkt erhoben und beruhen auf subjektiven Einschätzungen. Die abschließende Attribution zeigt, dass die meisten Teilnehmenden ihren Lernerfolg auf eine Kombination verschiedener Quellen zurückführen: neben dem KI-Mentor spielen Peer-Interaktionen und die Lehrperson eine zentrale Rolle.

6. Diskussion

6.1 Stabilisierungswirkung im Phasenmodell

Die Zusammenschau von Selbsteinschätzungen, offenen Antworten und Attributionen liefert Hinweise darauf, bei welchen Aufgaben der KI-Mentor als unterstützend erlebt wird. Der Beitrag des KI-Mentors wird dabei im Zusammenhang mit weiteren Unterstützungsquellen betrachtet. Am Messzeitpunkt 3 liegen die Mittelwerte von Wollen, Wissen und Können sowie des Stabilisierungsitems höher als am zweiten Messzeitpunkt. Die höheren Werte der drei Kerndimensionen beschreiben mehr Klarheit über Ziele und nächste Schritte sowie höheres Vertrauen in deren Umsetzung. Das ergänzende Stabilisierungsitem erfragt die Abstimmung zwischen Zielen, Wissen und Umsetzung (vgl. Kapitel 4.3). Sein höherer Mittelwert lässt sich auf Gruppenebene als Hinweis auf eine bessere wahrgenommene Passung und damit auf Stabilisierung im Sinne von Kapitel 2.1 deuten. Die berichteten technischen Schwierigkeiten widersprechen dieser Einschätzung nicht. Wie der KI-Mentor dazu beitrug, lässt sich erst anhand der offenen Antworten und Attributionen einordnen. Mit dem Tageswechsel und dem Übergang zur Durchführung überlagern sich ein möglicher Erholungseffekt, die weitere Projektarbeit und die Nutzung verschiedener Unterstützungsquellen. Ihr jeweiliger Beitrag zum Anstieg am Messzeitpunkt 3 lässt sich nicht bestimmen (vgl. Kapitel 6.5).

Am Messzeitpunkt 4 erreicht das selbsteingeschätzte Können seinen Höchstwert, während sich das Stabilisierungsitem kaum verändert. Zugleich wird der KI-Mentor häufiger nicht genutzt. Die begleitende Seminarbeobachtung zeigte, dass viele Gruppen nun unmittelbar in Scratch arbeiteten und offene Schritte anhand ihrer eigenen Planung umsetzten. Dies passt zu einer fortgeschrittenen und zunehmend selbstständigen Projektarbeit. Ob frühere Interaktionen mit dem KI-Mentor dazu beitrugen, bleibt offen. Die häufigere Nicht-Nutzung begrenzt zudem die Aussagekraft der Bewertungen zur Unterstützung.

6.2 Mechanismen der Stabilisierung

Die offenen Antworten liefern Hinweise auf drei Mechanismen und Bedingungen. Erstens wird der KI-Mentor als Hilfe bei der Konkretisierung unklarer Ziele beschrieben. Dies entspricht seiner dialogisch und fragend angelegten Gestaltung. Zweitens erzeugen seine Rückfragen ein Spannungsverhältnis zwischen Nutzersteuerung und didaktischer Führung. Teilnehmende empfanden die Beharrlichkeit teilweise als anstrengend oder einschränkend und versuchten, den Dialog anders auszurichten. Aus Sicht der Seminarleitung sollte sie Reflexion und Begründung einfordern und die Ausrichtung an den Seminarzielen sichern. Die wiederholten Rückfragen verweisen damit auf eine beabsichtigte Reibung, die unterschiedlich bewertet wurde. Drittens wird Unterstützung häufig mehreren Quellen zugeschrieben. Da Combination den KI-Mentor einschließen kann, aber nicht zwingend einschließt, erscheint er als mögliches Element eines Arrangements aus Peer-Unterstützung, Lehrbegleitung und weiteren KI-Werkzeugen.

Der KI-Mentor sollte die Studierenden beim Entwickeln eigener Lösungswege unterstützen. Die begleitende Seminarbeobachtung deutete darauf hin, dass Peers teilweise konkrete Probleme für andere Studierende lösten. Die Unterstützungszuschreibungen lassen offen, ob Hilfen den eigenen Lösungsprozess begleiteten oder einzelne Arbeitsschritte ersetzten.

6.3 Grenzen des Systems: Gedächtnis und Rigidität

Zwei wiederkehrende systembezogene Kritikpunkte lassen sich aus der Architektur erklären, werden dadurch aber nicht aufgehoben. Das wahrgenommene Gedächtnisproblem verweist auf den fehlenden Kontexttransfer zwischen den Modulen der untersuchten Implementierung. Ein gemeinsamer Kontextspeicher könnte bereits erarbeitete Informationen auch nach einem Modulwechsel verfügbar machen (vgl. Kapitel 3.5). Die eingeschränkte Flexibilität hängt teilweise mit didaktischen Regeln zusammen: Das Planning-Modul ist auf die Klärung der Planung vor der Implementierung ausgerichtet, das Didactics-Modul auf fokussierte Rückfragen. Diese beabsichtigte Führung wurde teilweise als Einschränkung erlebt: „The AI was very rigid in its answers and what it wanted you to do, it was hard to steer it from the 'assigned' task [...]“ (Z2,14).

6.4 Attribution und Wechsel zu generischer KI

Die Zuschreibungen an den Messzeitpunkten 3 und 4 zeigen, dass generische KI-Tools in der Umsetzungsphase eine relevante Unterstützungsquelle waren. Dafür sind zwei miteinander vereinbare Deutungen plausibel. Einerseits kann der Wechsel darauf hindeuten, dass die angebotene Unterstützung nicht zum Bedarf oder zu den Erwartungen der Studierenden passte. Der eingesetzte KI-Mentor konnte Programmabläufe beschreiben und passende Scratch-Blöcke empfehlen, aber keinen direkt importierbaren Scratch-Code bereitstellen. Diese technische Begrenzung war zugleich eine didaktische Entscheidungsgrundlage für die Wahl von Scratch: Die Studierenden sollten die Vorschläge selbst in Blockprogramme übertragen und deren Funktionsweise prüfen. Die in Kapitel 5.3 beschriebenen Schwierigkeiten zeigen, dass die begleitende Unterstützung bei konkreten Programmierfragen teilweise als unzureichend erlebt wurde. Andererseits kann er als aufgabenbezogene Auswahl von KI-

Tools verstanden werden. Eine bereits gewonnene Orientierung könnte es den Studierenden erleichtert haben, in der Umsetzungsphase gezielt auf passendere Anwendungen zurückzugreifen. Ob und in welchem Umfang der KI-Mentor zu dieser Orientierung beitrug, lässt sich anhand der vorliegenden Daten nicht bestimmen. Der Wechsel wäre in dieser Lesart kein Defizit, sondern Ausdruck einer zunehmend selbstständigen Entscheidung über die benötigte Unterstützung. Die Zuschreibungen an Peers und Lehrperson unterstreichen zudem die Bedeutung menschlicher Begleitung im Seminar. Der KI-Mentor ist entsprechend als ergänzendes Element eines didaktischen Arrangements zu verstehen, das fachlichen Austausch und persönliche Begleitung einschließt. In zwölf gültigen Antworten mit alleiniger Peer-Group-Attribution über alle vier Messzeitpunkte trifft eine niedrige Erwartungskonformität der Antworten des KI-Mentors (M = 2,17) mit einer hohen selbsteingeschätzten Handlungsfähigkeit zusammen (Können: M = 4,92). Ob die Peer-Unterstützung dazu beigetragen hat, bleibt offen.

6.5 Limitationen

Die Studie ist mit mehreren Einschränkungen verbunden. Erstens ist die Stichprobe mit 85 bereinigten Antworten für quantitative Aussagen begrenzt. Die Zahl gültiger Likert-Antworten sank von 27 am ersten auf 16 am vierten Messzeitpunkt. Zweitens beruhen die Likert-Daten auf Selbsteinschätzungen, die Aussagen über wahrgenommene Stabilisierung erlauben, nicht jedoch über objektivierte Lernzuwächse. Drittens ist die Doppelrolle des Autors als Seminarleiter und Forschender methodisch zu reflektieren. Die damit verbundene Nähe zum Untersuchungsfeld ist im Design-Based Research nicht ungewöhnlich, kann jedoch sozial erwünschte Antworten und eine selektive Interpretation begünstigen. Der KI-Mentor wurde als zu erprobender Prototyp vorgestellt und die Spiele wurden nicht benotet. Diese Rahmenbedingungen konnten kritische Rückmeldungen erleichtern. Da sich die anonymen Antworten nicht über die Messzeitpunkte hinweg denselben Personen zuordnen lassen, bilden die Erhebungen wiederholte Querschnitte. Unterschiede zwischen den Mittelwerten können auch allein auf eine veränderte Zusammensetzung der Antwortenden zurückgehen. Viertens erlaubt das Design keine kausalen Aussagen. Die beobachteten Veränderungen können auch auf Zeitdruck, die Limitationen von Scratch oder Gruppendynamiken zurückgeführt werden. Fünftens dient das Wollen-Wissen-Können-Dreieck als theoretischer Rahmen für die Interpretation der Selbsteinschätzungen. Die Aussagen zur Stabilisierung beziehen sich auf die von den Studierenden wahrgenommene Passung dieser Dimensionen im untersuchten Seminar. Die Befunde sind daher kontextbezogen zu interpretieren.

6.6 Implikationen für die Weiterentwicklung

Aus den Befunden ergeben sich vier Entwicklungsperspektiven. Erstens sollte ein gemeinsamer Kontextspeicher Informationen über Modulwechsel hinweg erhalten. Zweitens könnte der Dispatcher den Grad didaktischer Führung an Projektphase und Lernvoraussetzungen anpassen. Die begleitenden Beobachtungen der Seminarleitung legen nahe, dass die Programmiererfahrung für die Bewertung und den wahrgenommenen Nutzen des KI-Mentors bedeutsam sein könnte. Da diese Vorerfahrung nicht systematisch erhoben wurde, bleibt

der Zusammenhang offen. Weitere Untersuchungen sollten klären, welche Unterstützung Studierende ohne Programmiererfahrung benötigen und wie sich die als hilfreich wahrgenommenen Unterstützungsformen mit zunehmender Erfahrung verändern. Daraus könnten Hinweise entstehen, wie sich die didaktische Führung KI-gestützter Hilfswerkzeuge innerhalb des Seminarkonzepts an unterschiedliche Lernvoraussetzungen anpassen lässt. Das Verhältnis von Push- zu Pull-Interaktionen und die Scores aus dem Push-Modus kommen als Indikatoren infrage, deren Tragfähigkeit empirisch zu prüfen wäre.

Drittens sollte das Coding-Modul die Studierenden gezielter dabei unterstützen, geplante Programmabläufe in Scratch umzusetzen und Programmierfehler zu erkennen und zu beheben. Die gemeinsame Erkundung des KI-Mentors könnte beim Übergang zur Programmierung am Coding-Modul vertieft werden. Dabei sollten die Studierenden erproben, wie sie die Hinweise und Rückfragen des Mentors für eigene Lösungswege nutzen, diese in Blockprogramme übertragen und deren Funktionsweise prüfen können. Viertens könnte das Plenum nach jeder Arbeitsphase als dokumentierter Rückmeldeort in das Forschungsdesign aufgenommen werden. Es war in diesem Durchlauf nicht als Erhebungsinstrument angelegt und konnte aus zeitlichen Gründen nicht systematisch dokumentiert und ausgewertet werden. Eine solche Dokumentation könnte die Fragebogendaten ergänzen und in die Anpassung des folgenden DBR-Zyklus einfließen.

7. Fazit

Der Beitrag dokumentiert die Entwicklung und erste Evaluation eines modularen KI-Mentors für ein interdisziplinäres Serious-Game-Seminar. Die quantitativen Daten zeigen an den späteren Messzeitpunkten höhere mittlere Selbsteinschätzungen von Wollen, Wissen und Können. Aufgrund der nicht verknüpfbaren Antworten bleibt jedoch offen, inwieweit diese Unterschiede auf individuelle Veränderungen oder eine veränderte Zusammensetzung der Antwortenden zurückgehen. Die offenen Antworten liefern konkrete Hinweise darauf, dass der KI-Mentor bei Planungsaufgaben Orientierung bieten und die Klärung von Zielen und nächsten Schritten unterstützen kann. Die Unterstützung wird überwiegend dem Zusammenwirken verschiedener Quellen zugeschrieben. Der Rückgriff auf generische KI-Tools in der Umsetzungsphase lässt sich als Reaktion auf Grenzen des Coding-Moduls und als selbstständige Werkzeugwahl verstehen. Welchen Anteil frühere Orientierung durch den KI-Mentor daran hatte, bleibt offen. Entwicklungsbedarf besteht beim modulübergreifenden Gedächtnis, bei der Anpassung didaktischer Führung und bei der Coding-Unterstützung. Künftige Fallanalysen mit über die Messzeitpunkte hinweg verknüpfbaren Antworten und Angaben zur jeweiligen Arbeitsphase könnten genauer klären, bei welchen Aufgaben der KI-Mentor als hilfreich erlebt wird.

Der Beitrag verdeutlicht, wie technische Gestaltungsentscheidungen, didaktische Anforderungen und weitere Unterstützungsangebote in einem konkreten Einsatzszenario zusammenwirken. Für fachübergreifende Themenfelder wie informatische Grundbildung und Demokratiebildung kann modulares KI-Mentoring fachliche, didaktische, technische und planerische Anforderungen verbinden. Neben dem Sprachmodell prägen die didaktische Gestaltung und Einbettung die Qualität eines solchen Systems. Weitere Entwicklung und fachdidaktische

Forschung sollten klären, unter welchen Bedingungen solche Systeme Lernprozesse individualisieren und strukturieren sowie zur Orientierung und Handlungsfähigkeit von Lernenden beitragen.

Danksagung

Gefördert durch die Stiftung Innovation in der Hochschullehre

Der Autor dankt der Stiftung Innovation in der Hochschullehre für die Förderung des Projekts „Civic Education Technology Hacks“ (CivicEdTecH) im Rahmen der Ausschreibung „Freiraum 2023“. Die Förderung ermöglichte die Entwicklung und theoretische Fundierung des Seminarkonzepts, seine praktische Erprobung sowie die Entwicklung des KI-Mentors.

Kofinanziert von der Europäischen Union

Das Blended Intensive Programme wurde im Rahmen von Erasmus+ gefördert (Projektnummer: 2024-1-DE01-KA131-HED-000208164-1).

Der Autor dankt auch den beteiligten Partnerhochschulen für die Zusammenarbeit im Rahmen des BIP. Für dessen Organisation und Durchführung gilt der Dank den Mitarbeitenden des International Office der PH Ludwigsburg sowie Mareike Oesterle, Stefanie Schlosser, Julian Kugler und allen beteiligten Hilfskräften. Außerdem dankt er Robert Mondry für die Unterstützung bei der Fehlerbehebung und Feinjustierung des KI-Mentors sowie den Studierenden für ihre Mitwirkung am Seminar und an den Befragungen.

Literatur

Amanullah, Kashif/Bell, Tim (2019): Analysis of Progression of Scratch Users based on their Use of Elementary Patterns. In: 2019 14th International Conference on Computer Science & Education (ICCSE), Toronto, 19–21 August 2019. IEEE, pp. 573–578. doi.org/10.1109/ICCSE.2019.8845495.

Bloom, Benjamin S. (1984): The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. In: Educational Researcher, 13 (6), pp. 4–16. doi.org/10.3102/0013189X013006004.

Brennan, Karen/Resnick, Mitchel (2012): New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA), Vancouver, pp. 1–25. media.mit.edu/publications/new-frameworks-for-studying-and-assessing-the-development-of-computational-thinking/.

Gee, James Paul (2005): Learning by Design: Good Video Games as Learning Machines. In: E-Learning and Digital Media, 2 (1), pp. 5–16. doi.org/10.2304/elea.2005.2.1.5.

Geldreich, Katharina/Talbot, Mike/Hubwieser, Peter (2019): Aufgabe ist nicht gleich Aufgabe – Vielfältige Aufgabentypen bewusst in Scratch einsetzen. In: Pasternak, Arno (Hrsg.): Informatik für alle. 18. GI-Fachtagung Informatik und Schule. Bonn: Gesellschaft für Informatik (GI-Edition – Lecture Notes in Informatics, P-288), S. 181–190. doi.org/10.18420/infos2019-c3.

Hauk, Dennis/May, Michael (2025): Mehr als digitale Partizipation. Schulische Demokratiebildung in einer digital geprägten Welt. In: Achour, Sabine/Sieberkrob, Matthias/Pech, Detlef/Zelck, Johanna/Eberhard, Philip (Hrsg.): Handbuch Demokratiebildung und Fachdidaktik. Band 1:

Grundlagen und Querschnittsaufgaben. Frankfurt am Main: Wochenschau Verlag, S. 236–247. doi.org/10.46499/2154.3314.

Keller, Sebastian/Krafft, Maren/Fraser, Gordon/Walkinshaw, Neil/Otto, Korbinian/Sabitzer, Barbara (2019): Improving Scratch Programming with CRC-Card Design. In: Proceedings of the 14th Workshop in Primary and Secondary Computing Education (WiPSCE '19). New York: ACM, pp. 1–4. doi.org/10.1145/3361721.3362114.

Klein, Kerstin (2008): Lernen mit Projekten. In der Gruppe planen, durchführen, präsentieren. Mülheim an der Ruhr: Verlag an der Ruhr.

Kulik, James A./Fletcher, John D. (2016): Effectiveness of Intelligent Tutoring Systems: A Meta-Analytic Review. In: Review of Educational Research, 86 (1), pp. 42–78. doi.org/10.3102/0034654315581420.

Mayring, Philipp (2022): Qualitative Inhaltsanalyse. Grundlagen und Techniken. 13., überarbeitete Auflage. Weinheim/Basel: Beltz.

Reinhardt, Sibylle (2022): Politik-Didaktik. Handbuch für die Sekundarstufe I und II. 10., überarbeitete Auflage. Berlin: Cornelsen.

Reinmann, Gabi/Herzberg, Dominikus/Brase, Alexa (2024): Forschendes Entwerfen. Design-Based Research in der Hochschuldidaktik. Bielefeld: transcript Verlag (Zukunft der Hochschule, 7). doi.org/10.14361/9783839474242.

Sowa, Hubert (2011): Grundlagen der Kunstpädagogik – anthropologisch und hermeneutisch. Ludwigsburg: Verlag der Pädagogischen Hochschule (Schriften zur Kunstpädagogik Favorite, 5).

Spieler, Bernadette (2022): Gendersensible Gestaltung eines Computational-Thinking-Kurses mithilfe des PECC-Modells. In: Knackstedt, Ralf/Sander, Jürgen/Kolomitchouk, Jennifer (Hrsg.): Kompetenzmodelle für den Digitalen Wandel. Berlin/Heidelberg: Springer, S. 183–201. doi.org/10.1007/978-3-662-63673-2_9.

Süpfle, Rouven (2025): Opening the Blackbox: Game Design as a Bridge between Computer Science and Civic Education in Teacher Training. In: ICERI2025 Proceedings. Valencia: IATED Academy, pp. 6632–6640. doi.org/10.21125/iceri.2025.1824.

Thorgeirsson, Sverrir/Weidmann, Theo B./Su, Zhendong (2026): Computer Science Achievement and Writing Skills Predict Vibe Coding Proficiency. In: Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26). New York: ACM, Article 947, pp. 1–17. doi.org/10.1145/3772318.3791666.

Wing, Jeannette M. (2006): Computational Thinking. In: Communications of the ACM, 49 (3), pp. 33–35. doi.org/10.1145/1118178.1118215.

Informationen zum Autor



Rouven Süpfle ist Realschullehrer mit den studierten Fächern Deutsch, Geschichte und Politikwissenschaft. Seit dem Schuljahr 2026/2027 unterrichtet er an der Riedbachschule, einer Gemeinschaftsschule in Steinheim an der Murr. Zuvor war er zwei Jahre zur Koordination und Durchführung des Projekts „Civic Education Technology Hacks“ an die Pädagogische Hochschule Ludwigsburg abgeordnet. Dort ist er weiterhin als Lehrbeauftragter in der Abteilung Politikwissenschaft tätig. Über mehrere Jahre unterrichtete er zudem Informatik an Realschulen und besuchte entsprechende Fortbildungen. Bereits vor seiner Abordnung führte er fächerübergreifende Projekte mit Schülerinnen und Schülern durch.

In seinem Dissertationsvorhaben untersucht er, wie sich Demokratiebildung und informatische Grundbildung durch die Entwicklung digitaler Lernspiele in der Lehrkräftebildung verbinden lassen. Aus dieser Arbeit hat sich die KI-gestützte Lernbegleitung als weiterer Arbeitsschwerpunkt entwickelt.

rouven.suepfle@ph-ludwigsburg.de
r.suepfle@mailbox.org

Zitationshinweis:
Süpfle, Rouven (2026): Zwischen Wollen, Wissen und Können: Ein modularer KI-Mentor als adaptive Stabilisierungsinstanz in projektbasierten Lernprozessen der Hochschullehre. In: *Ludwigsburger Beiträge zur Medienpädagogik – LBzM*, 26, S. 1–25. doi.org/10.21240/lbzm/26/17.

Anhang A: Systemprompts des modularen KI-Mentors

Wiedergegeben sind die Klassifikationsinstanz sowie das Modul Planning in beiden Interaktionsmodi, wortgetreu und ungekürzt. Die Prompts wurden im System auf Englisch eingesetzt und bleiben im Original. Planning veranschaulicht exemplarisch, wie die Unterstützung bei Planungsaufgaben gestaltet ist.

A.1 Klassifikationsinstanz (Dispatcher)

You are the Classifier of the Civic Education Mentor.

Classify each user message into exactly one TAG and one MODE.

TAGS (choose one):

PLANNING – projects, timelines, milestones, structure.

DIDACTICS – pedagogy, learning goals, methods, assessment, reflection.

DESIGN – game mechanics, storytelling, UX/UI, visuals.

CODING – programming, debugging, logic, syntax, technical implementation.

MODE (choose one):

PULL – user asks for explanations, help, ideas or guidance.

PUSH – user requests a challenge OR is answering a tutor question.

MODE priority rules (in order):

1) Sticky PUSH:

If the previous assistant message asked a question or gave an assignment, keep MODE = PUSH.

2) Answer rule:

If the message looks like an answer (reasoning, justification, solution, short reply like yes/no, listed points, code snippet), choose PUSH.

3) Explicit challenge:

If the user asks to be tested or challenged (quiz, prüf mich, Aufgabe, challenge, üben, test), choose PUSH.

4) Otherwise:

Choose PULL only if the user clearly seeks explanation/help and is not answering.

TAG rules:

Choose the most concrete level.

If code is present → CODING.

If pedagogy/learning theory → DIDACTICS.

If game design/UX → DESIGN.

If structure/timeline → PLANNING.

Context:

Retain previous TAG and MODE by default.

Change only if the domain clearly shifts.

Never switch from PUSH to PULL if the user is answering.

Output strictly:

TAG: <CLASS>, MODE: <MODE>

No additional text.

A.2.1 Planning im Pull-Modus (Wissensaufbau)

You are the Planning Expert for Scratch-based Civic Education game projects in a tightly scheduled university setting.

Respond in precise C1-level English. Use short, clear sentences. Avoid repetition and filler language.

Core rules:

- No coding before architecture is defined.

- If topic, competence focus, or conflict structure are unclear, require clarification before planning implementation.

- Do not compensate missing didactic clarity with scheduling advice.

Project constraints:

- Implementation in Scratch.

- Limited coding sessions (Tuesday, Wednesday).

- Mandatory research phase and academic documentation.

- AI-generated assets must be time-budgeted.

- "Frog and his swamp" is an optional reference model.

Architectural requirements:

1) Define all sprites and responsibilities (CRC logic).

2) Define interaction and broadcast structure.

3) Plan phases: research, concept, architecture, prototype, implementation, testing.

4) Ensure realistic scope and extensibility (e.g. multilingual support).

Interaction style:

- Ask architecture-first questions.

- Stop premature implementation.

- Highlight structural risks early.

- Focus strictly on planning.

Goal:

Ensure a feasible, well-structured Scratch project that prevents later rework.

A.2.2 Planning im Push-Modus (Wissensabsicherung)

You are the Planning Tutor for Scratch-based Civic Education Scratch projects.

Respond in precise C1-level English. Use short, clear sentences. Be direct and analytical.

Non-negotiable rule: No implementation without defined architecture.

Project constraints:

- Scratch-based game.

- Limited coding sessions (Tuesday, Wednesday).

- Mandatory research and documentation.

- AI assets must be time-budgeted.

- "Frog and his swamp" is optional.

Evaluation criteria:

1) Clear sprite list with responsibilities.

2) Explicit interaction and messaging structure.

3) Separation of logic and content.

4) Consideration of future changes.

5) Realistic scope and milestones.

6) Defined research phase.

Interaction rules:

- Ask one focused architecture question at a time.

- Identify structural gaps immediately.

- Require concrete decisions.

- Explain long-term consequences of weak design.

After feedback, end with:

[SCORE]=<number>

Do not add text after the score.

A.3 Übrige Module

Didactics, Design und Coding sind ebenso aufgebaut. Die Push-Instanzen schließen jeweils mit der Ausgabe [SCORE]=<number>, die das System ausliest und protokolliert; die Pull-Instanzen geben keinen Score aus.