
Jahrbuch Medienpädagogik 17:
Lernen mit und über Medien in einer digitalen Welt
Herausgegeben von Klaus Rummler, Ilka Koppel, Sandra Aßmann,
Patrick Bettinger und Karsten D. Wolf

Die ersten 1000: Computational Thinking als obligatorische Ausbildung für Primarschullehrpersonen in der Schweiz

Alexander Repenning, Nora Anna Escherle und Anna Lamprou

Zusammenfassung

Die Umsetzung des Lehrplans 21, der mit dem Modul «Medien und Informatik» informatische Bildung (IB) in die Schweizer Primarschulen bringen wird, wirft wichtige Fragen auf: Insbesondere fragt sich, wie die Umsetzung gegen die bei vielen Primarlehrpersonen ablehnende Haltung gegenüber Informatische Bildung gelingen kann. Um angehende Primarlehrpersonen mit dem erforderlichen Fachwissen und den didaktischen Kompetenzen auszustatten, führte die PH FHNW im Herbst 2017 das zweiteilige Pflichtmodul «Informatische Bildung» für Studierende des Bachelors Primarstufe ein. Zudem fokussierte das Modul darauf, in den Studierenden eine nachhaltige, positive Handlungsveränderung hinsichtlich Informatische Bildung auszulösen. Die Erstdurchführung wurde durch mehrfache Datenerhebungen wissenschaftlich begleitet. Eine zentrale Fragestellung war, inwiefern der Besuch des Moduls die Haltungen der Studierenden bezüglich ihres Selbstkonzepts hinsichtlich Informatische Bildung und der Relevanz von Informatischer Bildung auf der Primarstufe beeinflusste. Dieser Artikel präsentiert kurz das didaktische Konzept des Moduls und beschreibt eine Auswertung der Daten, die bei der ersten Durchführung der beiden Modulteile erhoben wurden.

This work is licensed under a Creative Commons
Attribution 4.0 International License
<http://creativecommons.org/licenses/by/4.0/>



1000 Teachers trained: Computational Thinking as mandatory Education of Preservice Elementary School Teachers in Switzerland

Abstract

The mandatory implementation of the Swiss national curriculum, the "Lehrplan 21" including computer science education, raises an important question: How can a course overcome common negative dispositions of elementary school preservice teachers towards computer science? In 2017 the School of Education FHNW (PH FHNW) introduced a Scalable Game Design-based module consisting of a science course followed by a didactics course. This module is mandatory for all elementary school pre-service teachers. To assess the efficacy of this module an instrument consisting of a 14 questions questionnaire was used to compute effect sizes. In the first two cohorts about 1000 teachers were trained in four states in Switzerland. Initially, Scalable Game Design was developed for the professional development of in-service teachers in the US. The goal of this research is to explore if the core concept of Scalable Game Design – acquiring Computational Thinking through the design of games and simulations – could be effectively transferred from in-service teacher development to pre-service teacher education.

1. Einleitung

Die Allgegenwart der Digitalisierung in allen Lebensbereichen und deren beträchtlicher Einfluss ist mittlerweile unbestritten. Zwar ist es deshalb nicht notwendig, dass alle Menschen Informatikerinnen oder Informatiker oder gar Programmiererinnen oder Programmierer werden, um sich in diesem Kosmos der digitalen Transformation zurechtzufinden. Es ist jedoch unabdingbar, dass jede Person möglichst früh die notwendigen Grundkompetenzen erwirbt, die es ihr ermöglichen, zu verstehen, wie diese digitalisierte Informationsgesellschaft funktioniert, und die sie befähigen, sich darin nicht nur zurechtzufinden, sondern auch aktiv daran mitzuwirken. Dementsprechend wichtig sind entsprechende Anpassungen der Inhalte und auch der Strukturen im Bildungssystem. Ein wichtiger

bildungspolitischer Schritt in eben diese Richtung erfolgte mit der Erarbeitung des Lehrplans 21 (LP21), dessen Einführung mittlerweile von allen deutschsprachigen Kantonen beschlossen wurde.

Zwischen der Einführung des Lehrplans und dessen fachkundiger Umsetzung liegen jedoch Welten. Das langfristige Ziel muss eine Ausbildung für Primarlehrpersonen sein, in welcher Informatische Bildung eine zentrale Rolle spielt. Denn nur, wenn Studierende an Pädagogischen Hochschulen selbst eine substanzielle fachwissenschaftliche und fachdidaktische Ausbildung in den relevanten Kompetenzbereichen erhalten, können sie zukünftig als Primarlehrpersonen fachkundige Vermittler eben jener Kompetenzen sein. Als eine der ersten Schweizer Pädagogischen Hochschulen hat die Pädagogische Hochschule der Fachhochschule Nordwestschweiz (PH FHNW) Informatische Bildung in die Ausbildung zukünftiger Primarschullehrpersonen integriert. Die Kurskonzepte sind ausführlich beschrieben worden (Repenning et al. 2018). Nach der abgeschlossenen Ausbildung von mittlerweile schon über 1000 Lehrpersonen gibt dieser Beitrag eine Übersicht über die Kurse und fokussiert sich dann auf eine Auswertung der Wirksamkeit der Kurse.

2. Das Modul «Informatische Bildung»

Das Modul besteht aus zwei, aufeinander aufbauenden Teilen, welche die Studierenden in ihrem ersten Studienjahr absolvieren müssen: Der erste Teil «Fachwissenschaft IB» (FWIB) findet jeweils im Herbstsemester, der zweite Teil «Fachdidaktik IB» (FDIB) im Frühjahrssemester statt. Beide Module sind wöchentlich stattfindende zweistündige Präsenzveranstaltungen über 14 Wochen, die den Studierenden im Fall des Bestehens (mind. Note 4 – befriedigend) mit 2 ECTS kreditiert werden. Bei der Entwicklung des Moduls stellte sich die zentrale Frage, was Informatische Bildung auf der Primarstufe sein müsse und welche Kompetenzen Lehrpersonen benötigen, um dies zu vermitteln. Zwar definiert der LP21 Medien und Informatik als eigene, fächerübergreifende Kompetenzbereiche und beschreibt zahlreiche zu vermittelnde Kompetenzen. Doch bietet er weder ein ganzheitliches didaktisches Konzept noch konkrete Umsetzungsideen für die Lehre.

Informatische Bildung kann als Kombination von Informatik, IKT-Anwendungskompetenz, und Medienbildung verstanden werden (Felderer und Breu 2018). IKT-Anwendungskompetenzen, wie beispielsweise die Fähigkeit Microsoft Word zu benutzen um einen Bericht zu schreiben, werden an der PH FHNW aber nicht als Teil des Moduls Informatische Bildung vermittelt. Es wird angenommen, dass grundlegende Anwendungskompetenzen schon vorher an Gymnasien erworben wurden. Es hat sich mittlerweile jedoch gezeigt, dass sich diese Annahme leider häufig als falsch erwiesen hat. Als Backupstrategie werden einzelne Anwendungskompetenzen in anderen Fachbereichen eingebettet. So werden z.B. Kenntnisse von Word in den Deutschunterricht integriert, oder Kenntnisse von Excel in den Mathematikunterricht eingebettet. Wichtig ist daneben die Medienbildung mit dem Ziel u.a. medienpädagogische Kompetenzen zu erwerben (Blömeke 2005). Mit dem stark eingeschränkten Zeitgefäß muss sich das Modul Informatische Bildung allerdings momentan auf mediendidaktische Kompetenzen fokussieren. So lernen die Studierenden beispielsweise als Teil des Fachdidaktik IB Kurses, wie man *Zones of Proximal Flow Tutorials* (Basawapatna, Repenning, und Savignano 2019) konstruiert und evaluiert (Repenning et al. 2019). Im Vordergrund stehen dabei didaktische Aspekte wie z.B., was sind wirksame Videoinstruktionen, oder wie bietet man differenziertes Scaffolding (Hmelo-Silver, Duncan, und Chinn 2007) an, und nicht Anwendungskompetenzen wie z.B., wie benutzt man YouTube oder PowerPoint. Es ist geplant, mit der nächsten Akkreditierung (2024) eigene Zeitgefäße für Medienbildung zu erzeugen.

Mit dem Informatik-Teil macht das Modul Informatische Bildung eine klare programmatische Setzung: Es fasst Informatische Bildung primär als konzeptuelle Erarbeitung und fächerübergreifende Anwendung von Computational Thinking – jenes analytisch-strategische *Denken mit dem Computer*, welches die Qualitäten des Menschen mit den Fähigkeiten des Computers kreativ vereint, um komplexe Probleme zu lösen (Wing 2006). Computational Thinking (CT) ist somit eine wichtige transversale Grundkompetenz, die eine mündige Teilhabe in der heutigen digitalisierten Informationsgesellschaft ermöglicht. Aus diesem Grund spielt Computational Thinking eine zentrale Rolle bei der Vermittlung aller Inhalte, Konzepte und Methoden des Moduls Informatische Bildung und ist oberstes Kompetenzziel.

2.1 Computational Thinking

Die Definition von CT, die dem Modul Informatische Bildung zugrunde liegt, basiert primär auf derjenigen der Informatikerin Jeannette Wing. Sie fasst Computational Thinking als einen Gedankenprozess, der sowohl die Formulierung eines Problems als auch die Repräsentation der Problemlösung so darstellt, dass sie von Menschen oder durch Maschinen ausgeführt werden könnte (Wing 2014). Nach Wing ist Computational Thinking eine für alle Menschen grundlegende Kompetenz, die möglichst früh an jeder Schule und integriert in andere Fächer gelehrt werden sollte (Wing 2006). Bei dieser Aussage muss allerdings verstanden werden, dass Computational Thinking nicht gleichgesetzt werden sollte mit Programmieren oder Informatik. Computational Thinking schliesst auch wichtige soziale und ethische medienpädagogische Aspekte ein. Der in den USA populäre *Advanced Placement Computer Science Principles* Kurs (CollegeBoard 2017b), beispielsweise, beschreibt die Entwicklung einer kooperativen und ethischen «computing culture» als grundlegendes Ziel.

Obgleich es bis heute keine allgemeingültige Definition von Computational Thinking gibt, herrscht auf dem Gebiet der Informatischen Bildung weitgehend Einigkeit sowohl über die wichtigsten Elemente dieser Kompetenz als auch über deren eminente Bedeutung für alle, die an unserer zunehmend digitalisierten Welt aktiv mitwirken möchten. Dies ist eine erfreuliche und vielversprechende Ausgangslage für die baldige feste Verankerung von Computational Thinking als zu vermittelnde Kernkompetenz auch und vor allem in Schweizer Schulen. Kompetenzen im Medien- und Informatikunterricht – massgebend für die Ausbildung von Lehrpersonen in der Deutschschweiz – sind im Lehrplan 21 (Deutschschweizer Erziehungsdirektoren-Konferenz – D-EDK 2014) beschrieben. Diese Kompetenzen nehmen zwar keinen expliziten Bezug auf Computational Thinking als benannten Begriff, sind jedoch kompatibel mit den Zielen von Computational Thinking, inklusive dem unten beschriebenen Computational Thinking Prozess.

Die dem Modul Informatische Bildung zugrundeliegende Definition orientiert sich stark an den Bedürfnissen der Primarschule, ihren Lehrpersonen und ihren Schülerinnen und Schülern. In diesem Sinne wird Computational Thinking nach Seymour Papert als handlungsorientiertes,

transversales Denken mit dem Computer definiert (Papert 1996). Dabei fungiert der Computer als Hilfsmittel, das den Denkprozess unterstützt und die Konsequenzen des eigenen Denkens visualisiert. Computational Thinking bedeutet, ein Problem so zu formulieren, dass eine computer-gestützte Lösung vollzogen werden kann. Gemäss Wings Definition kann der CT-Prozess in drei Phasen gegliedert werden (Abb. 1).

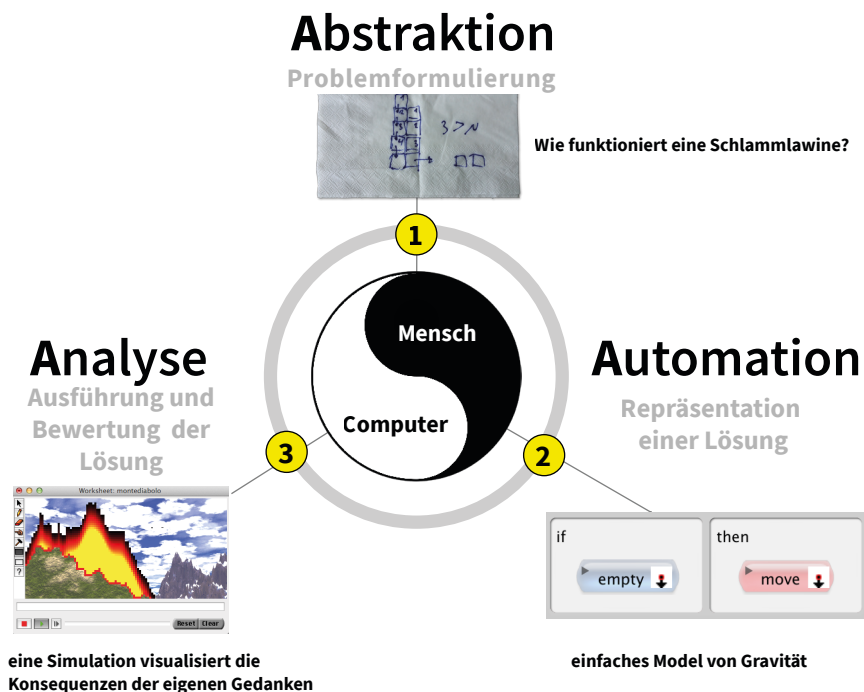


Abb. 1.: Computational Thinking als iterativer Prozess von Abstraktion, Automation und Analyse.

- **Abstraktion (Problemformulierung):** In der einfachsten Form die Formulierung einer präzisen Fragestellung basierend auf einer Problemanalyse.
- **Automation (Repräsentation einer Lösung):** Die exakte, unzweideutige Darstellung eines Lösungsweges basierend auf einer Kombination von Text und Diagrammen, zum Beispiel in Form eines Computerprogramms.

- **Analyse (Ausführung und Bewertung der Lösungsrepräsentation):**
Nach Ausführung eines Lösungsweges, beispielsweise in Form eines Computerprogramms, erfolgt die Evaluation von dessen Qualität.

Die zentrale Vision der beiden Veranstaltungen des Moduls Informatische Bildung ist es, zukünftige Primarlehrpersonen in der Kernkompetenz des Computational Thinking zu schulen und sie dazu zu befähigen, diese Kernkompetenz in ihren Schülerinnen und Schülern aufzubauen und zu fördern.

2.2 Die drei Säulen des Moduls

Um diese zentrale Vision zu verwirklichen, behandelt das Modul Informatische Bildung den Kompetenzbereich Informatik mit den Kompetenzzielen hinsichtlich Datenstrukturen, Algorithmen und Informationssystemen prioritär. Diese Setzung beruht wesentlich auf zwei eng verknüpften Faktoren: Zum einen erwartete man von den Studierenden in diesem Bereich wenig Vorwissen und ging davon aus, dass dieses zunächst aufgebaut werden muss. Zum anderen ist aber die Informatik ein hoch anspruchsvolles Fachgebiet, in welchem innerhalb von zwei Semestern nur rudimentäre Kompetenzen aufgebaut werden können. Dazu gehört nebst grundlegenden Programmierkenntnissen und Informatikwissen primär das Verständnis, was kreative Prozesse ausmacht und wie diese nachhaltig angeregt werden können. Eine geeignete Strategie zur Vermittlung von Computational Thinking kombiniert projektorientierten Unterricht mit einem Fokus auf Motivation und Lernstrategien, nutzt stufengerechte und anwendungsfreundliche Programmier- und Kreativitätswerkzeuge und ist strukturiert auf der Basis von allgemein als fundamental anerkannten Informatik-Prinzipien. In Abstimmung auf das oberste Kompetenzziel des Computational Thinking und dessen Verknüpfung mit den Kompetenzen des LP21 basiert das Kurskonzept des Moduls Informatische Bildung auf drei Säulen:

1. **Die Motivations- und Lernstrategie von Scalable Game Design (SGD).**
SGD stellt Motivation ins Zentrum des Lernprozesses. Dieser wird begleitet durch das Kreieren von Spielen und Simulationen und deren

fortlaufende Anpassung an neue Ideen und Kompetenzniveaus. SGD ist eine erprobte Motivationsstrategie, Informatik auf spannende Weise einem breiten Publikum zugänglich zu machen (Repenning et al. 2015). Zugleich beinhaltet es die *Zones of Proximal Flow*-Lernstrategie (Basawapatna et al. 2013), die es Lernenden ermöglicht, neue Kompetenzen durch auf ihre individuellen Niveaus angepasste und dadurch motivierende Herausforderungen zu erwerben. *Zones of Proximal Flow* (ZPF) vereinigt Wygotski's lerntheoretisches Konzept von sozialem Lernen, Zone der proximalen Entwicklung (ZPD) (Vygotsky 1978), mit Csikszentmihalyi's psychologischen Konzept von Flow (Csikszentmihalyi 1990). Mittels Aufgaben aus dem Bereich des Spiel- und Simulationsdesigns von SGD erlangen Studierende des Moduls Informatische Bildung wichtige CT- und Programmierkompetenzen des LP21 sowie zahlreiche Anregungen für deren Vermittlung.

2. **Stufengerechte und kontextspezifisch einsetzbare Computational Thinking Tools.** Diese unterstützen Lernende gezielt darin, die kognitiven und affektiven Herausforderungen des CT-Prozesses zu meistern (Repenning 2017). Die Studierenden erhalten einen Überblick zu Computational Thinking Tools wie AgentCubes online, die sich für die Lehre von Computational Thinking auf der Primarstufe und die Vermittlung der LP21-Kompetenzen im Bereich Medien und Informatik eignen, und werden umfassend in deren Anwendung und Einsatz im Unterricht geschult.
3. **Die sieben grossen Themen der Informatik** – in Analogie zu den «7 Big Ideas» des «AP Computer Science Principles»-Kurses (CollegeBoard 2017a). Die Themen Kreativität, Abstraktion, Daten, Algorithmen, Programmieren, das Internet und globale Auswirkungen umfassen die wichtigsten Prinzipien und Konzepte der Informatik und erläutern deren gesellschaftliche Konsequenzen. Insbesondere für den ersten, fachwissenschaftlichen Teil des Moduls liefern sie die relevanten Inhalte und diverse konzeptionelle Aspekte, die im Einklang mit den Kompetenzen des LP21 gelehrt werden.

Bei der Ausarbeitung des Moduls Informatische Bildung wurden zwei Aspekte als notwendig erachtet, um Computational Thinking sowie die

Fachkompetenzen des Lehrplanmoduls «Medien und Informatik» professionell unterrichten zu können: Zum Einen bedarf es breit abgestützter fachlicher Kompetenz im Bereich der Informatik, welche die Fähigkeit des Computational Thinking mit der Kenntnis wichtiger Konzepte, Inhalte und Methoden der Informatik verbindet. Zum anderen erfordert ein professioneller Unterricht *informatik-didaktische* Kompetenzen, denn diese befähigen die Lehrperson, Lehr- und Lernprozesse gestützt auf wissenschaftliche Erkenntnisse und reflektierte Erfahrungen zu planen und zu gestalten. Entsprechend der zweifachen Zielsetzung des Studienfaches werden die Bereiche Fachwissenschaft und Fachdidaktik unterschieden.

2.2.1 Fachwissenschaft

Der erste Modulteil FWIB fokussiert auf die Entwicklung grundlegender Kompetenzen bezüglich Inhalte, Konzepte und Methoden der Informatik. Durch handlungsorientierte Lernaufgaben fördert er dabei von Beginn an die Befähigung der Studierenden zur eigenständigen und kreativen Anwendung von Computational Thinking als essenzielle transversale Grundkompetenz. Thematisch bilden die 7 *Big Ideas* der Informatik in Feinabstimmung auf die Kompetenzziele des LP21 das Grundgerüst der Veranstaltung. Während jeweils zwei Doppellektionen erhalten die Studierenden fundierte Einführungen und handlungsorientierte Lernaufgaben zu jedem der sieben Themen. Dies soll es den Studierenden ermöglichen, in diesen Bereichen die notwendigen Fachkompetenzen aufzubauen, um als Primarschul-Lehrpersonen ihre Schülerinnen und Schüler in informatischer Bildung fachkundig zu begleiten. Für den schrittweisen und nachhaltigen Aufbau der Grundkompetenz des Computational Thinking ist es zudem notwendig, dass Studierende mit Hilfe von Computational Thinking Tools, wie z.B., AgentCubes (Abb. 2), Grundkenntnisse im Programmieren erwerben. AgentCubes ist frei nutzbar in der Schweiz. Um dies zu gewährleisten, räumt die Veranstaltung dem Erstellen von digitalen Artefakten in Form von Computerspielen und -simulationen grossen Raum ein. Nach ersten Gehversuchen ohne Computer mit Computer Science Unplugged-Aktivitäten erstellen die Studierenden bald einfache Computerspiele und -simulationen anhand konkreter Vorgaben. Die dabei erworbenen spezifischen

Kompetenzen und die fortwährend geschulte Anwendung des Computational Thinking befähigen sie schliesslich, eigenständig ein individuelles Programmierprojekt zu planen und durchzuführen.



Abb. 2.: AgentCubes ist ein Computational Thinking Tool mit dem 2D und 3D Spiele und Simulationen gebaut werden können.

Die Computational Thinking-Kompetenzen die nötig sind, um Spiele und Simulationen zu bauen werden aufgebaut durch sogenannte Computational Thinking Patterns (Koh et al. 2014). Diese Patterns verkörpern Abstraktionen (Abb. 1, Abstraktion) von fundamentalen Objektinteraktionen wie das kollidieren von zwei Objekten (z.B. die Kollision von einem Auto mit dem Frosch im Frogger Spiel), generieren, absorbieren, ziehen, stossen, verfolgen und andere Interaktionen. Die Studierenden lernen diese Computational Thinking Patterns umzusetzen (Abb. 1, Automation) und können damit nicht nur die geforderten Kompetenzen des Lehrplan 21 in der Schweiz abdecken, sondern sind auch in der Lage aufwendige Spiele und Simulationen zu bauen. Abb. 3 zeigt einige Beispiele. Das Spektrum reicht von 1980 Arcade-style-Games wie Donkey Kong, Logik Spiele wie Tetris, Ökosysteme, Fantasy Spiele, und 3D Spiele/Simulationen.

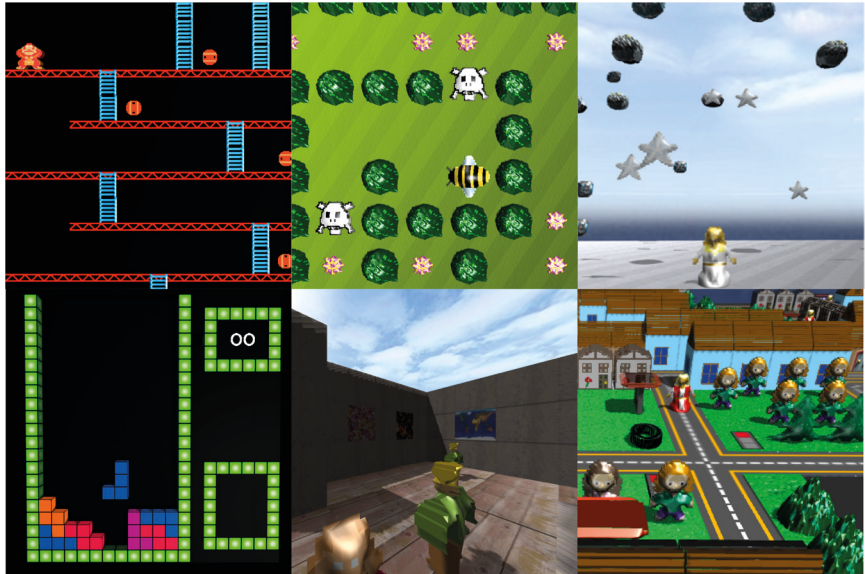


Abb. 3.: Beispiele von Spielen die von Studierenden gebaut worden sind.

2.2.2 Fachdidaktik

Der zweite Modulteil FDIB übersetzt die im ersten Modulteil behandelten Inhalte und Konzepte für den Unterricht mit Primarschülerinnen und -schüler. Im Zentrum stehen dabei die didaktischen Konzepte von SGD für die Vermittlung Informatischer Bildung im Sinne von CT. SGD implementiert auf praxisnahe und projektorientierte Weise die *Zones of Proximal Flow*-Theorie (Basawapatna et al. 2013; Repenning et al. 2015). Diese Theorie kombiniert bildungsrelevante Aspekte der *Flow*-Theorie (Csikszentmihalyi 1990) mit denen der *Zone of Proximal Development*-Theorie (Vygotsky 1978) und bereichert diese wiederum mit didaktischen Prinzipien von Papert (1996) wie beispielsweise den Konstruktivismus. Das ursprüngliche Flow-Framework von Csikszentmihalyi beschreibt die Relation von Anforderungen und Fähigkeiten als drei mögliche psychologische Zustände: Überforderung (Anforderungen > Fähigkeiten), Flow (Anforderungen \approx Fähigkeiten), und Unterforderung (Anforderungen < Fähigkeiten).

Aus einer allgemeindidaktischen Perspektive werden die in der Fachwissenschaft verwendeten Methoden für den Einsatz auf der Primarstufe übersetzt. Wie in der Fachwissenschaft spielt das Erstellen von digitalen

Artefakten in Form von Computerspielen und -simulationen eine zentrale Rolle. In der Fachdidaktik rückt jedoch die Frage nach geeigneten Formen der Lehre von Computational Thinking für den Lernprozess von Primarschülerinnen und -schülern in den Fokus. Da es noch wenig Lehrmittel im Bereich Informatischer Bildung gibt, macht die Fachdidaktik die Studierenden nicht nur mit didaktischer Theorie vertraut, sondern stattet sie zudem mit zahlreichen Ideen aus, wie man Informatische Bildung konkret unterrichten kann – und zwar integriert in andere Fächer. Dies entspricht dem InformatikPLUS-Ansatz, der als wichtiger Aspekt der Primarschuldidaktik im Bereich Informatischer Bildung eingeführt wird. Das bedeutet, dass IB-Kenntnisse und -Kompetenzen im Zusammenspiel mit Kompetenzen aus thematisch geeigneten anderen Fachbereichen wie beispielsweise Natur-Mensch-Gesellschaft (NMG), Musik oder Mathematik aufgebaut werden.

3. Empirische Evaluation

Bei diesem Paper handelt es sich um einen Erfahrungsbericht mit einer Forschungsfrage. Die Forschungsfrage evaluiert die Wirksamkeit von Scalable Game Design als pädagogische Strategie:

Forschungsfrage mit dem Ziel Computational Thinking durch Game Design zu vermitteln: wie wirksam, gemessen als Effektgrösse (Hill et al. 2008), ist die pädagogische Strategie Scalable Game Design, welche ursprünglich für Weiterbildung konzipiert wurde, in der Ausbildung?

Im Gegensatz zu den typischen Teilnehmenden an Weiterbildungen müssen die Teilnehmenden der Ausbildung Kurse zur Informatischen Bildung zu belegen (Lamprou und Repenning 2018) auch, wenn sie nicht an dem Thema interessiert sind. Scalable Game Design ist ursprünglich für erfahrene Lehrpersonen entwickelt worden, welche ihr eigenes Interesse zur Informatik gut abschätzen können (Repenning et al. 2015). Im Gegensatz dazu ist es vielen zukünftigen Lehrpersonen in der Ausbildung nicht klar, was genau Informatische Bildung bedeutet. Eine informelle Befragung hat gezeigt, dass ein Grossteil dieser zukünftigen Lehrpersonen diese Programmierkurse nicht freiwillig belegt hätte.

Design-Based Implementation Research (Fishman et al. 2013) wurde als Forschungsansatz verwendet um einerseits die Wirksamkeit von Scalable Game Design als pädagogische Strategie zu untersuchen, und andererseits die beiden Veranstaltungen graduell, mit Bezug auf die Zufriedenheit der Lernenden, zu verbessern. Design Based Research (DBR) (Wang und Hanafin 2005) und Design Based Implementation Research (DBIR) (Fishman et al. 2013) sind designbasierte Forschungsansätze mit dem Ziel effektive, skalierbare und nachhaltige Bildungskonzepte aufzubauen und systematisch zu erforschen. Anstatt zu versuchen all die verschiedenen Faktoren zu isolieren die das Lernen beeinflussen, wie mit standardisierter Experimentalforschung, benutzt DBR/DBIR authentische Feldstudien um den komplexen Systemcharakter von Lernsystemen zu untersuchen. Es hat sich gezeigt, dass DBR/DBIR Ansätze nicht nur gültige Ergebnisse liefern sondern oft wesentlich effektiver sind in der Umsetzung (Fishman et al. 2013). Traditionelle Ansätze basierend auf standardisierter Experimentalforschung leiden häufig darunter, dass sie zwar theoretisch interessante Ergebnisse liefern aber dann nur sehr aufwendig, oder sogar gar nicht, in der Praxis integriert werden können.

Der DBR/DBIR Ansatz ist auf dem Design von Lernprozessen – sogenannten Interventionen basiert, um spezifische Anforderungen, wie z.B. das Lernen von Programmieren, abzudecken. Forschende sowie Praktikerinnen und Praktiker arbeiten Hand in Hand am Design der Interventionen, der Durchführung der Interventionen, d.h. die Umsetzung des Designs und der Analyse mit gemeinsam ausgewählten Forschungsinstrumenten. Dieser Prozess wird mehrfach iteriert um Resultate der Analyse in einem Re-Design einfließen zu lassen und die Intervention zu wiederholen. Beide, DBR sowie DBIR Ansätze, produzieren theoretische sowie praktische Forschungsergebnisse. Im Vergleich zu DBR hat aber DBIR einen ausgeprägteren Fokus auf Implementation. Das heisst, das Ziel von DBIR ist es vor Allem nachhaltige Verschiebungen der Praxis zu bewirken.

Die Wirksamkeit von Scalable Game Design wurde mit Effektgrössenanalysen basierend auf Pre/Post Studien untersucht. Im Gegensatz zu standardisierter Experimentalforschung in welchen Effekte statistische Unterschiede zwischen unabhängigen Behandlungs- und Kontrollgruppen beschreiben werden in DBIR oft Effektgrössen berechnet basierend auf Pre/Post Daten der gleichen Gruppe.

Die hier präsentierten Ergebnisse basieren auf insgesamt vier Datenerhebungen, die im Rahmen der Präsenzveranstaltungen der beiden Moduleile FWIB (Herbstsemester 2017) und FDIB (Frühjahrssemester 2018) vorgenommen wurden. In beiden Modulteilen erfolgten jeweils zwei Datenerhebungen: Eine erhob die Ausgangsdaten in der jeweils ersten Sitzung des Semesters und die andere erhob die Post-Daten in der letzten Sitzung. Die anwesenden Studierenden wurden gebeten, auf freiwilliger und anonymer Basis, an ihren eignen Computern einen Online-Fragebogen auszufüllen. Ein Grossteil der Fragen in Ausgangsfragebogen und Post-Fragebogen waren identisch, allerdings enthielten die Post-Fragebögen zusätzliche Fragen zur Kursevaluation. Die Fragebögen bestanden aus diversen Fragen mit Fragen zu Demographie, Haltungen gegenüber Informatik und Informatische Bildung sowie zahlreichen Selbstwirksamkeitsfragen im Zusammenhang mit Technologie im Allgemeinen und Informatik im Besonderen. Alle Fragen verwendeten eine fünfstufige Likert-Skala (1 = Trifft nicht zu -- 5 = Trifft zu). Den ersten Modulteil FWIB besuchten 581 Studierende des Bachelors Primarstufe. Von diesen füllten im 541 Studierende den FWIB-Ausgangsfragebogen und 474 füllten den FWIB-Post-Fragebogen aus. Von den Ausfüllenden des Ausgangsfragebogens waren 379 weiblich (74.02%) und 133 männlich (25.98%). Sie befanden sich grösstenteils im 1. und 3. Semester und waren zwischen 18 und 23 Jahren alt. Den zweiten Modulteil FDIB besuchten insgesamt 423 Studierende. Von diesen füllten 403 den FDIB-Ausgangsfragebogen und 311 den FDIB-Post-Fragebogen aus. Erneut waren zwei Drittel der Ausfüllenden des FDIB-Ausgangsfragebogens weiblich (75.76%), ein Drittel männlich (24.24%) und ihr Alter betrug grösstenteils zwischen 19 und 23 Jahren.

4. Ergebnisse

Die folgenden Abschnitte stellen ausgewählte Ergebnisse im Detail vor. Da dieser Beitrag das Modul Informatische Bildung nicht als Ganzes in den Blick nimmt, werden hier nur die Daten des ersten Fragebogens (FWIB-Pre) und die des vierten und letzten Fragebogens (FDIB-Post) präsentiert und zueinander in Beziehung gesetzt. Zudem präsentiert dieser Beitrag lediglich Auswertungsergebnisse zu den 14 Fragen, die im direkten

Zusammenhang stehen mit der anfangs formulierten Fragestellung zu Haltungen und Selbstwirksamkeitserfahrungen der Studierenden hinsichtlich Informatik und IB. Da die Datenerhebung anonym war und es deutliche Fluktuationen hinsichtlich der Anzahl und der Zusammensetzung der Teilnehmenden gab, waren wir nicht in der Lage, die Ausgangsdaten und die Post-Daten als verbundenen Datensatz zu behandeln. Stattdessen wurde ein einfacher t-Test durchgeführt, um herauszufinden, ob es signifikante Unterschiede zwischen der Einschätzung der verschiedenen Fragen vor und nach dem Besuch des Moduls Informatische Bildung gibt. Die Daten für die Berechnungen stammen aus dem FDIB-Post-Fragebogen. Als Vergleichswerte für die Berechnungen dienen die arithmetischen Mittel aus dem FWIB-Ausgangsfragebogen. Die statistische Signifikanz wurde anhand des Benjamini-Hochberg-Verfahrens berechnet (Benjamini und Hochberg 1995). Die folgende Tabelle präsentiert die arithmetischen Mittelwerte (M), Standardabweichungen (SD), Effektgrößen (Cohens d) und statistischen Signifikanzen (p-Werte) für 14 Fragen des FWIB-Pre-Fragebogens und des FDIB-Post-Fragebogens.

Fragen	FWIB-Pre	FDIB-Post	Effektgröße, p-Wert
1. Ich kann gut mit Computern umgehen	M=3.06 SD=0.85	M=3.55 SD=0.83	d = 0.58 p < .001
2. Ich kann programmieren	M=1.3 SD=0.64	M=2.98 SD=0.88	d = 2.18 p < .001
3. Ich würde gerne mehr Wissen über Computer und Informatik erlernen.	M=3.23 SD=1.2	M=3.32 SD=1.07	d = 0.08 p = n.s.
4. Ich fände es cool, ein Computerspiel zu entwickeln	M=3.2 SD=1.36	M=3.07 SD=1.25	d = -0.10 p = n.s.
5. Ich glaube, dass Informatik schwierig ist	M=3.64 SD=0.98	M=2.63 SD=0.97	d = -1.04 p < .001
6. Ich denke, dass Informatik langweilig ist	M=2.31 SD=1.02	M=2.43 SD=1.01	d = 0.12 p = n.s.
7. Ich finde, Computerspiele zu entwickeln ist eine hervorragende Form, Informatik zu lehren	M=3.46 SD=1.06	M=3.51 SD=1.03	d = 0.05 p = n.s.
8. Ich glaube, dass Informatik für meinen Beruf als Primarlehrperson wichtig ist	M=3.78 SD=1	M=3.78 SD=0.95	d = 0 p = n.s.

Fragen	FWIB-Pre	FDIB-Post	Effektgrösse, p-Wert
9. Ich glaube, dass Informatik für die Zukunft meiner Schüler*innen wichtig ist	M=4.24 SD=0.86	M=4.26 SD=0.8	d = 0.02 p = n.s.
10. Ich fürchte, Eltern könnten denken, dass ich nicht genug über Computer und Technologie weiss, um ihre Kinder auf die Zukunft vorzubereiten	M=1.83 SD=0.98	M=1.94 SD=0.97	d = 0.11 p = n.s.
11. Wenn ich eine technologische Herausforderung antreffe, gebe ich schnell auf	M=2.74 SD=1.09	M=2.69 SD=1.08	d = -0.05 p = n.s.
12. Ich werde wahrscheinlich nie wirklich gut sein mit Technologie	M=2.53 SD=1.11	M=2.38 SD=1.03	d = -0.14 p < .05
13. Ich denke, meine Fähigkeiten werden nie über das Benutzen der gängigsten Programme (Word, E-Mail) hinausgehen	M=2.26 SD=1.11	M=2.05 SD=1.03	d = -0.2 p < .001
14. Die Idee, dass Schule in Zukunft stärker durch Technologie geprägt wird, löst in mir Angst oder Ablehnung aus	M=1.99 SD=1.03	M=2.08 SD=1.02	d = 0.09 p = n.s.

Tab. 1.: Ausgewählte Ergebnisse des ersten (FWIB-Pre) und des letzten Fragebogens (FDIB-Post).

4.1 Deskriptive Statistik

Zu Beginn schätzten die Teilnehmenden ihre Computerkenntnisse als durchschnittlich ein ($M=3.06$) und gaben an, dass sie nicht programmieren können ($M=1.3$). Sie bekundeten ein mittelmässiges Interesse daran, programmieren zu lernen ($M=3.23$) oder ein Computerspiel zu entwickeln ($M=3.2$). Zwar gaben sie an zu denken, das Fach Informatik sei eher schwierig ($M=3.64$), sehen es aber nicht als langweilig ($M=2.31$) und erachten das Entwickeln von Computerspielen tendenziell als eine gute Idee, um Informatik zu lernen ($M=3.46$). Sie bekundeten die Haltung, Informatik sei eher wichtig für ihren zukünftigen Beruf der Primarlehrperson ($M=3.78$) und sehr wichtig für die Zukunft ihrer Schülerinnen und Schüler ($M=4.24$). Die Mittelwerte der fünf Fragen, welche die Selbstwirksamkeit der Teilnehmenden hinsichtlich Technologie im Allgemeinen und Computern im Besonderen messen (Fragen 10-14), zeugen durchweg von einer tendenziell positiven Haltung aber nur kleiner Veränderungen.

4.2 Unterschiede Pre-Post

Die grössten, statistisch signifikanten Veränderungen mit starken bis sehr starken Effektgrössen erfolgten bei Fragen 1, 2 und 5: Am Ende des Moduls Informatische Bildung schätzten die Teilnehmenden ihre Kompetenzen im Umgang mit Computern als deutlich besser ein ($M=3.55$; $d=0.54$), waren viel eher der Ansicht programmieren zu können ($M=2.98$; $d=2.18$) und hielten Informatik für weit weniger schwierig ($M=2.63$; $d=-1.04$) als zu Beginn. Weitere signifikante Verbesserungen, wenn auch mit schwachen Effektgrössen, zeigen sich bei Fragen 12 und 13 zum Thema Selbstwirksamkeit: Die Teilnehmenden waren am Ende noch weniger der Meinung als zu Beginn, dass sie mit Technologie nie gut sein werden ($M=2.38$; $d=-0.14$) und dass ihre Fähigkeiten nie über Anwendungskenntnisse hinausgehen werden ($M=2.05$; $d=-0.2$). Alle anderen Fragen weisen keine signifikanten Unterschiede auf zwischen den Mittelwerten zu Beginn des Moduls Informatische Bildung und jenen am Ende.

5. Diskussion

Die Mittelwerte der Fragen 1-2 des ersten Fragebogens FWIB-Pre bestätigen, was die Modul-IB-Verantwortlichen vermutet hatten: Die Studierenden halten ihre Computerkenntnisse für eher durchschnittlich und haben praktisch keine Programmierkenntnisse. Dies wird allerdings kontrastiert von zahlreichen unerwartet positiven Ergebnissen: Auf Basis der Antworten kann man sagen, dass die Haltung der Studierenden gegenüber Informatik (Fragen 5, 6) weit weniger negativ war als erwartet. Auch die Haltung gegenüber dem Erwerb von Kompetenzen in Informatik im Allgemeinen und anhand der Entwicklung von Computerspielen im Besonderen (Fragen 3, 4, 7) war positiver als ursprünglich angenommen. Dies impliziert, dass obligatorischen IB-Kurse mit deutlich weniger Skepsis oder gar Widerwillen gesehen werden als die Entwicklerinnen und Entwickler des Moduls Informatische Bildung vermutet hatten. Ein wichtiger Grund für diesen erfreulich geringen Widerwillen mag sein, dass die Studierenden Informatik als sehr wichtiges Fach für die Zukunft ihrer Schülerinnen und Schüler erachten und dem Fach auch für ihren eigenen Beruf eine tendenziell wichtige Rolle zuweisen (Fragen 8-9).

Andererseits könnte man diese Werte auch als das Fehlen einer kritischen Haltung interpretieren. In diesem Zusammenhang ist es erfreulich, dass die Studierenden bei den Fragen 10-14 zur Selbstwirksamkeit bezüglich Technologie und Computern tendenziell günstige Selbsteinschätzungen abgaben. Dies gilt insbesondere für die Fragen, die sich auf die Selbsteinschätzung ihrer technologischen Kompetenzen im Rahmen des zukünftigen Berufsumfelds beziehen (Fragen 10, 14). Insgesamt war also die Ausgangslage deutlich positiver und von weit weniger starker Abwehrhaltung seitens der Studierenden geprägt als die Modulverantwortlichen befürchtet hatten. Zudem lassen die Ergebnisse des Vergleichs zwischen den Daten des ersten und des letzten Fragebogens vermuten, dass das Modul Informatische Bildung entscheidend zur Verbesserung einiger Werte beitragen konnte. Dies gilt insbesondere für die beiden Fragen (1-2) bezüglich der Computerkenntnisse. Die starke Effektgrösse, die für Frage 1 «Ich kann gut mit Computern umgehen» berechnet wurde ($d=0.54$), deutet an, dass die Studierenden denken, ihre Computerkenntnisse seien deutlich gewachsen. Dies gilt umso mehr für Frage 2 «Ich kann programmieren», für welches eine sehr grosse Effektgrösse berechnet wurde ($d=2.18$). Dies deutet darauf hin, dass die Studierenden aufgrund ihres Besuchs des Moduls Informatische Bildung einen signifikanten Zuwachs an Programmierkenntnissen erfuhren oder zumindest annahmen. Die Wahrnehmung der Studierenden deckt sich mit den von ihnen erbrachten Leistungen und Leistungsfortschritten während des Moduls Informatische Bildung – zum Einen in einer Programmier-Prüfung (durchgeführt am Ende des ersten Semesters) und zum Anderen in diversen Programmier-Projekten, die sie in beiden Semestern einreichen mussten.

Bezüglich der Haltung gegenüber Informatik (Fragen 5-6) impliziert die sehr starke Effektgrösse ($d=-1.04$) von Frage 5 an, dass der Besuch des Moduls Informatische Bildung insbesondere dazu beitrug, dass Teilnehmende *Informatik für weniger schwierig* hielten. Ähnlich positive Auswirkungen des Moduls IB, wenn auch mit deutlich schwächer ausgeprägten Effektgrössen, lassen sich für Frage 12 «Ich werde wahrscheinlich nie wirklich gut sein mit Technologie» ($d=-0.14$) und Frage 13 «Ich denke, meine Fähigkeiten werden nie über das Benutzen der gängigsten Programme (Word, Email) hinausgehen» ($d=-0.2$) annehmen. Es scheint, als haben die

Teilnehmenden durch den Besuch des Moduls Informatische Bildung einen Zuwachs an Selbstvertrauen in ihre technologischen und Computerbezogenen Kompetenzen erworben.

Effektgrößen können schwierig zu interpretieren sein. Cohen (1992) hat eine operationale Definition eingeführt. Werte von $[0.2, 0.5]$ sind kleine Effekte, $[0.5, 0.8]$ sind mittlere Effekte und Werte grösser als 0.8 sind grosse Effekte. Sawilowsky hat diese Quantifizierung erweitert (Sawilowsky 2009). Er würde die Effektgrösse für «Kann ich programmieren» mit $d=2.18$ als riesig ($d > 2$) bezeichnen. Hill et al. (2008) gehen noch weiter mit der Interpretation von Effektgrößen mit der Idee von empirischen Benchmarks. Ein Benchmark ist z.B. die Veränderung von Lesekompetenzen von Schülerinnen und Schülern von einem Jahr zum nächsten. Lese- und Schreibkompetenzen sowie das Erlernen von Programmieren, also die Kompetenz Programme zu lesen und zu schreiben, könnte man als «literacy» verstehen (diSessa 2000). Es ist nachvollziehbar, dass die Effektgrößen beim Erwerben einer Literacy jedes Jahr abnehmen. Von der 9. Zur 10. Klasse ist die Effektgrösse beim Lesen nur noch 0.19. Hingegen, vor dem Beginn der 1. Klasse bis zum Ende der 1. Klasse 1.52. Damit könnte man sagen, dass die Studierenden nach nur 14 Wochen eine selbst eingeschätzte Verbesserung mit einem Effekt bezüglich Programmierkompetenzen haben, der bedeutend grösser ist als die erworbenen Lesekompetenzen von Schülerinnen und Schüler in der ersten Klasse.

6. Fazit

Computational Thinking durch Scalable Game Design vermittelt hat sich im Rahmen der Ausbildung von Lehrpersonen als wirksame Methode erwiesen. Scalable Game Design basiert auf der Verwendung von Game Design basierten Aktivitäten. Die Demographie der Teilnehmenden an Aus- und Weiterbildungen ist grundsätzlich unterschiedlich. Die Daten zeigen, dass in der Ausbildung von zukünftigen Primarschullehrpersonen nur eine kleine Minderheit Vorerfahrungen im Programmieren hat. Eine riesige Effektgrösse von $d=2.18$ (Cohen) für die Selbsteinschätzung eigener Programmierkompetenz deckt sich mit der Qualität von programmierten Spielen und anderen Leistungsnachweisen. Zwar haben sich negative

Einstellungen von einigen skeptischen Studierenden gegenüber der Benutzung von Game Design für den Informatikunterricht durch den Kurs kaum verändert, aber dem gegenüber hat sich der Glaube, dass Informatik schwierig ist, stark reduziert.

Literatur

- Basawapatna, Ashok R., Alexander Repenning, Kyu Han Koh, und Hilarie Nickerson. 2013. «The Zones of Proximal Flow: Guiding Students through a Space of Computational Thinking Skills and Challenges». In *ICER '13 Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 67. ACM Press. <https://doi.org/10.1145/2493394.2493404>.
- Basawapatna, Ashok, Alexander Repenning, und Mark Savignano. 2019. «The Zones of Proximal Flow Tutorial: Designing Computational Thinking Cliffhangers». In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 428–34. Minneapolis MN USA: ACM. <https://doi.org/10.1145/3287324.3287361>.
- Benjamini, Yoav, und Yosef Hochberg. 1995. «Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing». *Journal of the Royal Statistical Society: Series B (Methodological)* 57 (1): 289–300. <https://doi.org/10.1111/j.2517-6161.1995.tb02031.x>.
- Blömeke, Sigrid. 2005. «Medienpädagogische Kompetenz. Theoretische Grundlagen und erste Befunde». In *Kompetenzdiagnostik: Theorien und Methoden zur Erfassung und Bewertung von beruflichen Kompetenzen*, herausgegeben von Andreas Frey, Reinhold S. Jäger, und Ursula Renold, 1. Aufl., 76–97. Landau: Verlag Empirische Pädagogik.
- Cohen, Jacob. 1992. «A Power Primer». *Psychological Bulletin* 112 (1): 155–59. <https://doi.org/10.1037/0033-2909.112.1.155>.
- CollegeBoard. 2017. «AP Computer Science Principles.» *The College Board*. <https://apstudent.collegeboard.org/apcourse/ap-computer-science-principles/course-details>.
- Csikszentmihalyi, Mihaly. 1990. *Flow: The Psychology of Optimal Experience*. New York: Harper Collins Publishers.
- Deutschschweizer Erziehungsdirektoren-Konferenz (D-EDK), Hrsg. 2014. «Lehrplan 21: Medien und Informatik». D-EDK. http://v-ef.lehrplan.ch/container/V_EF_DE_Modul_MI.pdf.
- diSessa, Andrea. 2000. *Changing Minds: Computers, Learning, and Literacy*. Cambridge, MA: The MIT Press.

- Felderer, Michael, und Ruth Breu. 2018. «Von Autofahrern und Autobauern – Zur Rolle der Informatik in der (Medien)bildung». In *Medienpädagogik. Herausforderungen für Lernen und Bildung im Medienzeitalter*, herausgegeben von Theo Hug, 77–83. MEDIEN – WISSEN – BILDUNG. Innsbruck: innsbruck university press. https://www.uibk.ac.at/iup/buch_pdfs/9783903187306.pdf.
- Fishman, Barry J., William R. Penuel, Anna-Ruth Allen, Britte Haugan Cheng, und Nora Sabelli. 2013. «Design-Based Implementation Research: An Emerging Model for Transforming the Relationship of Research and Practice». *Yearbook of the National Society for the Study of Education* 112 (2): 136–56. http://isls-naples.psy.lmu.de/video-resources/guided-tour/15-minutes-penuel/fishman_penuel.pdf.
- Hill, Carolyn J., Howard S. Bloom, Alison Rebeck Black, und Mark W. Lipsey. 2008. «Empirical Benchmarks for Interpreting Effect Sizes in Research». *Child Development Perspectives* 2 (3): 172–77. <https://doi.org/10.1111/j.1750-8606.2008.00061.x>.
- Hmelo-Silver, Cindy E., Ravit Golan Duncan, und Clark A. Chinn. 2007. «Scaffolding and Achievement in Problem-Based and Inquiry Learning: A Response to Kirschner, Sweller, and Clark (2006)». *Educational Psychologist* 42 (2): 99–107. <https://doi.org/10.1080/00461520701263368>.
- Koh, Kyu Han, Hilarie Nickerson, Ashok Basawapatna, und Alexander Repenning. 2014. «Early Validation of Computational Thinking Pattern Analysis». In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education - ITiCSE '14*, 213–18. Uppsala, Sweden: ACM Press. <https://doi.org/10.1145/2591708.2591724>.
- Lamprou, Anna, und Alexander Repenning. 2018. «Teaching How to Teach Computational Thinking». In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE 2018*, 69–74. Larnaca, Cyprus: ACM Press. <https://doi.org/10.1145/3197091.3197120>.
- Papert, Seymour. 1996. «An Exploration in the Space of Mathematics Educations». *International Journal of Computers for Mathematical Learning* 1 (1): 95–123. <https://doi.org/10.1007/BF00191473>.
- Repenning, Alexander. 2017. «Moving Beyond Syntax: Lessons from 20 Years of Blocks Programming in AgentSheets». *Journal of Visual Languages and Sentient Systems* 3 (1): 68–91. <https://doi.org/10.18293/VLSS2017-010>.
- Repenning, Alexander, Anna Lamprou, Nicolas Fahrni, und Nora Escherle. 2018. «Scalable Game Design Switzerland». Herausgegeben von Torsten Brinda, Ira Diethelm, Sven Kommer, und Klaus Rummmler. *Medienpädagogik: Zeitschrift für Theorie und Praxis der Medienbildung* 33 (Medienpädagogik und Didaktik der Informatik): 27–52. <https://doi.org/10.21240/mpaed/33/2018.10.31.X>.
- Repenning, Alexander, Anna Lamprou, Serge Petralito, und Ashok Basawapatna. 2019. «Making Computer Science Education Mandatory: Exploring a Demographic Shift in Switzerland.» In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 422–28. Aberdeen Scotland Uk: ACM. <https://doi.org/10.1145/3304221.3319758>.

- Repenning, Alexander, Ryan Grover, Kris Gutierrez, Nadia Repenning, David C. Webb, Kyu Han Koh, Hilarie Nickerson, u. a. 2015. «Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation». *ACM Transactions on Computing Education* 15 (2): 1–31. <https://doi.org/10.1145/2700517>.
- Sawilowsky, Shlomo S. 2009. «New Effect Size Rules of Thumb». *Journal of Modern Applied Statistical Methods* 8 (2): 597–99. <https://doi.org/10.22237/jmasm/1257035100>.
- Vygotsky, Lev. S. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Edited by Michael Cole, Vera John-Steiner, Sylvia Scribner and Ellen Souberman. Harvard University Press.
- Wang, Feng, und Michael J. Hannafin. 2005. «Design-Based Research and Technology-Enhanced Learning Environments». *Educational Technology Research and Development* 53 (4): 5–23. <https://doi.org/10.1007/BF02504682>.
- Wing, Jeannette M. 2014. «Computational Thinking Benefits Society». *40th Anniversary Blog - Social Issues in Computing*, New York: Academic Press (blog). 10. Januar 2014. <http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>.
- Wing, Jeannette M. 2006. «Computational Thinking». *Communications of the ACM* 49 (3): 33–35. <https://doi.org/10.1145/1118178.1118215>.

Danksagung

Forschung und Entwicklung dieser Ausbildung in der Schweiz wurden von der Hasler Stiftung, dem Schweizerischen Nationalfonds und der Pädagogischen Hochschule FHNW unterstützt. Scalable Game Design und AgentCubes wurden in den USA von der National Science Foundation unterstützt.