
Themenheft Nr. 56: Making & more: gemeinsam Lernen gestalten.

Herausgegeben von Bernadette Spieler, Manuela Dahinden, Klaus Rummler
und Tobias M. Schifferle

Gelingensbedingungen für die affektive Förderung von Kindern durch einen Robotik-Making-Kurs

**Befunde zum Zusammenhang von Schwierigkeiten und Spass und deren
Diskussion aus motivationspsychologischer Sicht**

Luisa Greifenstein¹ , Ewald Wasmeier¹ , Ute Heuer¹  und Gordon Fraser¹ 

¹ Universität Passau

Zusammenfassung

Algorithmisches Denken wird vermehrt bereits in der Grundschule gefördert. Während diese aktuelle Entwicklung die Lehrkräfte insbesondere bei der Unterstützung der Schüler:innen vor Herausforderungen stellt, liegt eine besondere Chance in der Förderung von affektiven Komponenten wie Interesse und Motivation. Dazu werden häufig spielerische und handlungsorientierte Methoden und Aufgaben eingesetzt, wofür sich wiederum der Ansatz der Maker-Education besonders eignet. In diesem Beitrag wird daher ein Robotik-Making-Kurs beschrieben und werden erste empirische Befunde diskutiert. In dem Kurs baute jedes Kind in vier Stunden seinen eigenen Roboter zusammen, indem es selbsttätig schraubte, steckte, verleimte, lötete und verkabelte. In weiteren vier Stunden wurde der Roboter mithilfe einer kindgerechten Entwicklungsumgebung programmiert. Um Gelingensbedingungen für affektiv förderliche Unterstützungsmassnahmen abzuleiten, führten wir eine Studie mit 45 Kindern im Alter von neun bis elf Jahren durch. Dazu wurden die Schwierigkeiten der Kinder während des Zusammenbauens und des Programmierens notiert. Daraus konnten mehrere Kriterien abgeleitet werden, die zu häufigen Schwierigkeiten führen. Zudem wurde jeweils nach Abschluss des Bauens und des Programmierens der erlebte Spass als Vorstufe intrinsischer Motivation abgefragt. So konnte festgestellt werden, dass Schwierigkeiten, die die Autonomie oder das Kompetenzerleben einschränken, mit vermindertem Spass einhergehen. Abschliessend werden daher Handlungsempfehlungen diskutiert, die affektiv förderliche Making-Aktivitäten im Unterricht ermöglichen sollen.



**Promoting Affective Components of Children by a Maker Course on Robotics.
Findings on the Relation of Difficulties and Fun and Their Discussion from a
Motivational Perspective**

Abstract

Computational thinking is increasingly fostered in primary schools. While this current development can be challenging for teachers, especially regarding the support of students, there is a huge opportunity in promoting affective components such as interest and motivation. Playful and action-oriented methods and tasks are often used which is why the maker education approach is particularly suitable. This paper describes a making course on robotics and discusses initial empirical findings. During the course, all children built their own robot in four hours by screwing, plugging, gluing, soldering and wiring on their own. In another four hours, the robot was programmed with the help of a child-friendly development environment. In order to derive criteria for promoting children affectively, we conducted a study with 45 children aged nine to eleven years. The children's difficulties during building and programming the robot were noted. By this, several criteria could be derived that lead to frequent difficulties. In addition, the children were asked about their fun (as a preliminary stage of intrinsic motivation) after they had finished building and programming. By this, we found that difficulties that limit autonomy or competence are related to reduced fun. Practical recommendations that enable making activities in the classroom that promote affective components are discussed.

1. Einführung

Informatische Inhalte werden vermehrt bereits in der Grundschule verankert (Döbeli Honegger und Hielscher 2017; Heintz, Mannila, und Färnqvist 2016; Nenner und Bergner 2022). Dies stellt sowohl die Lehrkräftebildung (Döbeli Honegger und Hielscher 2017) als auch die Grundschullehrkräfte (Greifenstein, Graßl, und Fraser 2021; Sentance und Csizmadia 2017) vor grosse Herausforderungen. Gleichzeitig haben selbst (Fach-)Lehrkräfte an weiterführenden Schulen teilweise Schwierigkeiten mit der Unterstützung von Schüler:innen während des Unterrichts (Michaeli und Romeike 2019; Yadav et al. 2016). Die algorithmische Förderung von Grundschulkindern geht aber auch mit Chancen einher, insbesondere im affektiven Bereich (Greifenstein, Graßl, und Fraser 2021). Hierfür bieten sich spielerische und handlungsorientierte Lernaktivitäten an, die in der Grundschule häufig eingesetzt werden (Hailey et al. 2016) und für die sich der Ansatz der Maker-Education besonders eignet (Ingold und Maurer 2019). Aufgrund der genannten Chancen und Herausforderungen sollten Lehrkräfte unterstützt werden, Feedback zu geben und dieses

affektiv förderlich zu vermitteln. In diesem Beitrag werden daher ein Robotik-Making-Kurs und die den Lernenden zur Verfügung stehende Unterstützung vorgestellt sowie (1) aufgetretene Schwierigkeiten und (2) deren Zusammenhang mit dem erlebten Spass untersucht. Das häufige Auftreten bestimmter Schwierigkeiten lässt sich anhand der Notizen der Betreuenden auf die Mehrschrittigkeit der Problemlösung, die Komplexität des Problems und die Vorerfahrungen der Kinder zurückführen. Darüber hinaus zeigte sich, dass Schwierigkeiten, die die Autonomie oder das Kompetenzerleben einschränken, mit vermindertem Spass einhergehen. Positive Emotionen wie Spass sind jedoch notwendig, um für das Lernen hilfreiche intrinsische Motivation und Interesse zu entwickeln (Deci und Ryan 1993; Renninger 2009). Deswegen werden abschliessend Handlungsempfehlungen für einen affektiv förderlichen Unterricht durch die Ermöglichung von Autonomie und Kompetenzerleben abgeleitet.

2. Physical Computing in der Grundschule

2.1 Informatische Inhalte in der Grundschule

Bereits seit mehreren Jahren wird die Verankerung informatischer Inhalte in den Grundschullehrplänen gefordert. Aktuell fordert auch die Ständige Wissenschaftliche Kommission der Kultusministerkonferenz (SWK), dass «ausgewählte Aspekte der Informatik (neben der Medienkompetenz) für den Sachunterricht fest zu verankern sind» (Köller et al. 2022). In der Schweiz (Döbeli Honegger und Hielscher 2017) und in der Hälfte der deutschen Bundesländer sind informatische Inhalte bereits verankert (Nenner und Bergner 2022). Dabei wird insbesondere der Themenbereich Algorithmen und Programmierung thematisiert (ebd.), der auch einen Inhaltsbereich der «Kompetenzen für informatische Bildung im Primarbereich» darstellt (Best et al. 2019). Der vorliegende Kurs greift diesen Inhaltsbereich auf und setzt ihn anhand des Physical Computing um. Physical Computing ist dabei nicht nur eine Variante von Programmieraktivitäten, sondern kann zudem die Umsetzung und Reflexion von interaktiven Objekten mithilfe von Hard- und Software einschliessen (Przybylla und Romeike 2014). Die Orientierung am Produkt passt wiederum zu den Maximen der Making-Bewegung, die die Erstellung und Weiterentwicklung von Produkten fokussiert (Ingold und Maurer 2019).

2.2 *Physical Computing im Bildungskontext*

Physical Computing wird häufig als Variante zur Einführung in die Programmierung genutzt und eröffnet weitere Möglichkeiten wie den Einbezug des Konstruierens oder von Sensoren und Aktoren (Przybylla und Romeike 2014). Für das Physical Computing eignen sich insbesondere sogenannte Build-Roboter, die im Gegensatz zu Use-Robotern erst zusammengebaut werden müssen (Catlin et al. 2018). Bisherige Studien haben gezeigt, dass Physical Computing bzw. Programmierunterricht mit Robotern Lernende sowohl affektiv als auch kognitiv fördern kann. So bewerten Grundschullehrkräfte das Build-Robotik-Set WeDo als hilfreich für die Förderung des algorithmischen Denkens (Chalmers 2018) und schätzen die Robotik generell als sehr lernförderlich ein (Scaradozzi et al. 2015). Auch Studierende nutzen gerne Roboter als Werkzeuge, um ihr Verständnis hinsichtlich Programmierung, Aktoren und Sensoren zu verbessern (Aroca et al. 2012). In Bezug auf affektive Faktoren wird Robotern häufig ein hoher Spassfaktor zugeschrieben, der sich jedoch nicht immer gänzlich bestätigen lässt (Levy und Ben-Ari 2015). Ausserdem bevorzugen Studierende, selbst einen Roboter zu haben statt einen von der Universität auszuleihen (Aroca et al. 2012). In einer Untersuchung von 24 Veröffentlichungen konnte bestätigt werden, dass Roboter in der Primar- und Sekundarstufe häufig motivierend und kognitiv aktivierend wirken, insbesondere wenn sie handlungsorientiert und projektbasiert eingesetzt werden (Amo et al. 2020).

3. Theoretische Hintergründe zum Kursdesign

Im Rahmen des Robotik-Making-Kurses sollen die Kinder affektiv gefördert werden. Der Kurs orientiert sich daher (1) an Empfehlungen zur Umsetzung des Making-Ansatzes und (2) an motivationspsychologischen Überlegungen, die im Folgenden beschrieben werden.

3.1 *Making*

Beim Making im medienpädagogischen bzw. Bildungskontext geht es nicht nur um «Machen», sondern um «produktives Technikhandeln» und dessen Reflexion. Mithilfe analoger und digitaler Techniken sowie unterschiedlicher Materialien erstellen Lernende selbstgesteuert, spielerisch und handlungsorientiert individuelle Objekte oder Prototypen (Ingold und Maurer 2019). Häufig wird deshalb von einer Produktorientierung gesprochen, die jedoch nicht im Sinne einer perfekten Endleistung zu verstehen ist, sondern auf die iterative Verbesserung des Produkts fokussiert (ebd.), womit gleichzeitig eine Prozessorientierung gegeben ist. Bei den Lernenden zeigt sich diese Verknüpfung von Produkt und Prozess wie folgt (Blikstein 2013): Die Idee des finalen Produkts ist häufig der motivationale Ausgangspunkt. Während des

Prozesses treten jedoch verschiedene Schwierigkeiten auf, die angenommen und überwunden werden, wodurch sich häufig auch das angestrebte Produkt ändert. Hierin zeigt sich wiederum die Verschiebung von der reinen Produkt- zur Prozessorientierung.

Auch für das Erreichen von Lernzielen spielt der Prozess eine wichtige Rolle: Durch die thematische Offenheit im Making-Kontext können insbesondere grundlegende Kompetenzen wie Problemlösefähigkeit oder Media Literacy gefördert werden (Knaus und Schmidt 2020). Daher bietet sich auch die Verknüpfung mehrerer Fächer an, deren Möglichkeiten im Making-Ansatz auch von Lehrkräften besonders geschätzt werden (Spieler, Schifferle, und Dahinden 2022). Diese Interdisziplinarität ist jedoch häufig einer von mehreren Reibungspunkten bei der Umsetzung von Making in der Schule. Dies liegt an curricularen Vorgaben und der Organisation von Regelschulen (Ingold und Maurer 2019). Auch deshalb bietet sich Making für informelle Lerngelegenheiten und ausserschulische Lernorte an (Bunke-Emden 2020). In MakerSpaces, die prinzipiell auch in der Schule umsetzbar sind, werden den Lernenden verschiedene Materialien, Werkzeuge und Formen der Unterstützung zur Verfügung gestellt (Ingold und Maurer 2019). Diese Unterstützung kann unterschiedliche Formen annehmen: Durch eine kollaborative Bearbeitung unterstützen sich die Lernenden gegenseitig und die Lehrkraft oder die Betreuenden fungieren als beratende Lernbegleiter:innen (ebd.).

3.2 Spass, Interesse und Motivation

In kürzeren Kursen bzw. zu Beginn von Kursen bietet es sich an, Vorläufer von langfristiger intrinsischer Motivation und individuellem Interesse zu fördern und zu untersuchen. Für die anfängliche Förderung sind dabei vor allem positive Emotionen wichtig, aber auch das individuelle Interesse hängt neben Wissen und anhaltendem Engagement mit dem Erleben von positiven Gefühlen zusammen (Renninger 2009). Dies überschneidet sich wiederum damit, dass intrinsisch motivierte Personen etwas u.a. wegen des innewohnenden Spasses und der daraus resultierenden Zufriedenheit machen (Deci und Ryan 1993). Auch im Programmierunterricht hat sich Spass als wirksamer intrinsischer Motivator erwiesen (Riedo et al. 2013; Long 2007), weswegen wir von diesem Konstrukt ausgehen.

4. Kursdesign

Der Kurs besteht aus vier Einheiten à zwei Zeitstunden. In den ersten beiden Einheiten wird ein Roboter von Grund auf zusammengesetzt, in den folgenden beiden Einheiten wird er programmiert. Zu Beginn jeder Einheit wird den Kindern ein

Überblick über deren Lernziele gegeben. Durch die folgenden verschiedenen Unterstützungsmassnahmen soll den Kindern einerseits Kompetenzerleben, andererseits eine gewisse Autonomie ermöglicht werden.

4.1 Allgemeine Umsetzung und Unterstützung

Während der Einheiten konnten die Kinder jederzeit Fragen an die Betreuenden stellen oder sich bei anderen Kindern erkundigen, was ebenso wie die einzelnen Aufgaben in Partnerarbeit die soziale Eingebundenheit stärken kann (Deci und Ryan 1993). Wenn die Kinder zwar vor Herausforderungen standen, aber keine Hilfe einforderten, wurde kein Feedback gegeben. Dadurch wurde versucht, ihnen zu ermöglichen, das Problem selbst zu lösen und sie nicht aus einem möglichen Flow zu reißen. Wenn ein Kind um Hilfe bat, wurde es zudem gefragt «Willst du es selbst versuchen und ich erkläre es oder soll ich es für dich machen?». So kann das Kind selbst entscheiden, wie viel Hilfe es gerade benötigt, wodurch beispielsweise Mädchen häufiger selbst programmieren möchten als Jungen (Greifenstein et al. 2024).

4.2 Umsetzung und Unterstützung des Bauens

Das Zusammenbauen des Roboters erfolgte gemeinsam in 30 Schritten. Der Roboter besteht aus Holzteilen, die von der Kursleitung entworfen wurden und mithilfe eines CO₂-Lasers geschnitten wurden, verschiedenen Sensoren, Aktoren und Kabeln. Jedes Kind baute seinen eigenen Roboter zusammen. Abbildung 1 gibt an, wie oft die jeweilige Tätigkeit durchgeführt wurde. Dabei wurde mit verschiedenen Materialien und Werkzeugen gearbeitet. Nur das Heisskleben wurde von der Kursleitung übernommen, da nur eine Heissklebepistole zur Verfügung stand; da im Werkraum des Lernorts ausreichend Lötgeräte zur Verfügung stehen, konnte jedes Kind selbst löten (Abbildung 2).

Tätigkeiten während des Zusammenbauens

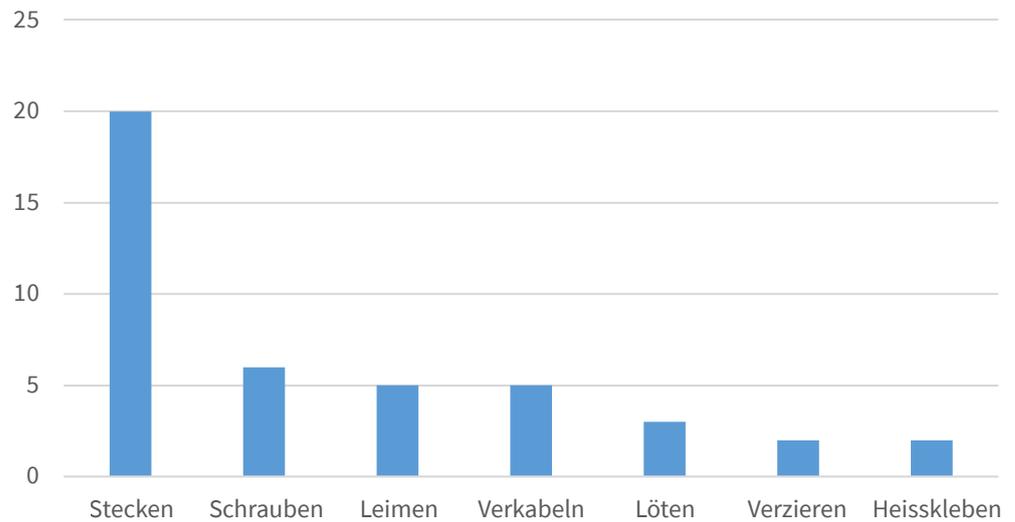


Abb. 1: Häufigkeit der jeweiligen Tätigkeiten während des Zusammenbauens.

Während das Grundgerüst für alle Kinder gleich ist, hatten die Kinder in einzelnen Schritten die Möglichkeit, ihren Roboter zu individualisieren, indem sie beispielsweise LEDs und einzelne Sensoren an passenden, aber selbst ausgewählten Stellen platzierten, einen Kopfschmuck auswählten und den Roboter bemalten (Abbildung 2). Die Interdisziplinarität des Kurses zeigt sich darin, dass neben den genannten handwerklichen Fähigkeiten und dem künstlerischen Geschick auch naturwissenschaftlich-technische Informationen einfließen: Während die kleinschrittige gemeinsame Umsetzung zwar die Selbststeuerung der Kinder reduzieren kann, ermöglicht sie der Kursleitung, allen Kindern grundlegende Hintergrundinformationen zu den verbauten Sensoren und Aktoren zu geben und gemeinsam mit den Kindern zu reflektieren. Zudem wurde die Aufgabe in der Präsentation anhand von Fotos grob erklärt, es wurden aber bewusst nicht alle Informationen vorweggenommen, um die Kinder kognitiv zu aktivieren. Im Sinne eines offenen Lernsettings könnte überlegt werden, die Informationen zur Aufgabe und zu den Bestandteilen des Roboters zu verschriftlichen oder zu vertonen, sodass die Kinder in ihrem eigenen Tempo arbeiten und Informationen bei Bedarf leichter differenziert werden können.

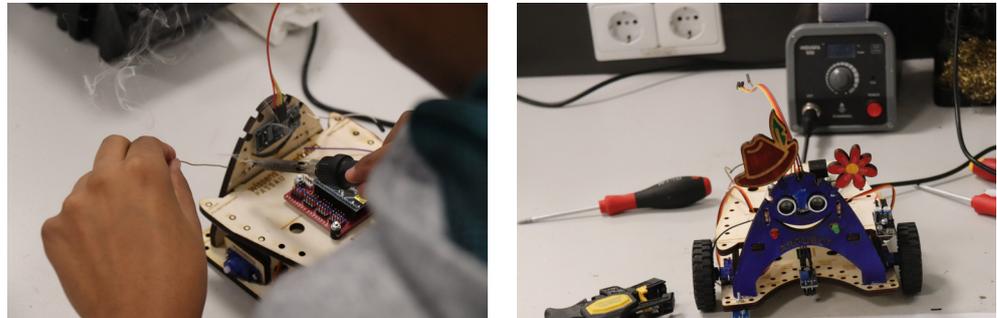


Abb. 2: Kind beim Löten der LED-Kabel und ein mit Kopfschmuck und Farbe verzierter Ardubot.

4.3 Umsetzung und Unterstützung des Programmierens

algorithmische Grundbausteine

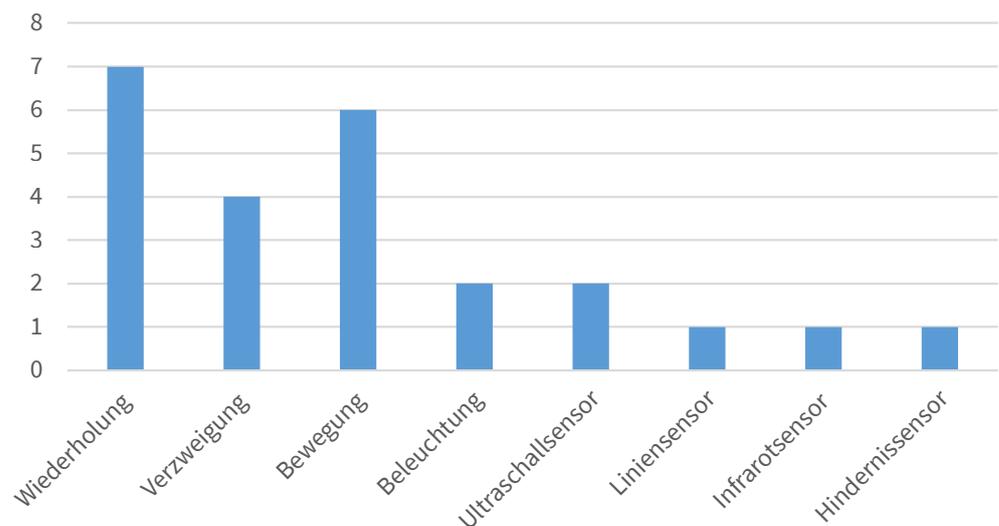


Abb. 3: Häufigkeit der jeweiligen Inhalte in den Programmieraufgaben.

In den insgesamt elf Programmieraufgaben werden die von der Gesellschaft für Informatik für die Grundschule empfohlenen algorithmischen Grundbausteine Anweisung, Sequenz, Wiederholung und Verzweigung (Best et al. 2019) eingeführt. Neben grundlegenden Komponenten des algorithmischen Denkens wie Abstraktion und Dekomposition (Selby und Woollard 2014) werden in den einzelnen Aufgaben verschiedene Konzepte ausprobiert. Abbildung 3 zeigt, welche Konzepte (algorithmische Bausteine/Aktoren/Sensoren) von den Kindern wie häufig verwendet wurden. Die blockbasierte Programmiersprache mBlock ermöglicht, farblich

kategorisierte Blöcke per Drag-and-Drop und ähnlich wie Puzzlesteine zusammenzufügen, was den Einstieg erleichtert. Allerdings können auch hier verschiedene fehlerhafte Codemuster programmiert werden (Obermüller et al. 2022; Frädriich et al. 2020).

Da einige Kinder zudem keine Vorerfahrung in der Programmierung hatten, wurde durch das Use-Modify-Create-Framework (Lee et al. 2011) ein Scaffolding ermöglicht: Wenn ein neues Konzept eingeführt wurde, bauten die Kinder zunächst den vorgegebenen Code nach und reflektierten diesen (= USE-Schritt), bevor sie dann Veränderungen an diesem Code vornahmen (= MODIFY-Schritt). Vereinzelt erstellten die Kinder auch Programme von Grund auf selbst (= CREATE-Schritt). Dieses Vorgehen wurde teilweise zusätzlich durch sogenannte Parsons Problems (Abbildung 4) unterstützt, bei denen die Lernenden Codeblöcke in die richtige Reihenfolge bringen (Du, Luxton-Reilly, und Denny 2020). Durch die Aufgaben konnten die Kinder einen ersten Einblick in die Programmierung gewinnen. Am Ende des Kurses freuten sich die Kinder, ihren eigenen Roboter mit nach Hause nehmen zu dürfen.

Diese Bausteine könnten dir bei dieser Aufgabe behilflich sein (manche Bausteine brauchst du mehrmals):

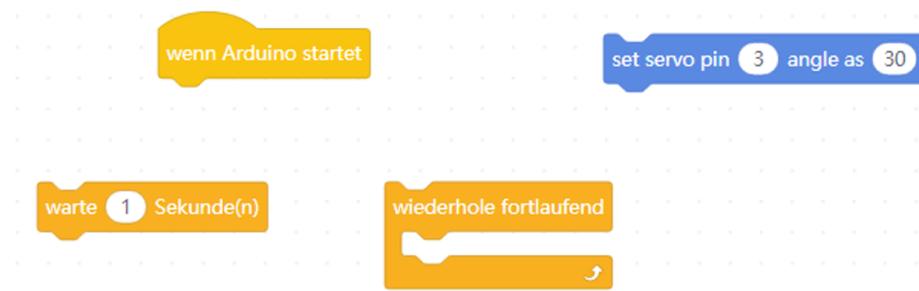


Abb. 4: Ein Parsons Problem (Du, Luxton-Reilly, und Denny 2020) für die Aufgabe des Fahrens einer Kurve mit zwischenzeitlichen Pausen.

5. Studiendesign

Die folgenden Forschungsfragen sollen mithilfe der vorliegenden Studie beantwortet werden:

- RQ 1: Inwieweit benötigen Kinder beim Bauen eines Roboters Feedback?
- RQ 2: Wie wirken sich die beim Bauen identifizierten Schwierigkeiten auf den Spass aus?
- RQ 3: Inwieweit benötigen Kinder beim Programmieren eines Roboters Feedback?
- RQ 4: Wie wirken sich die beim Programmieren identifizierten Schwierigkeiten auf den Spass aus?

5.1 Datensammlung

Der Kurs wurde vier Mal durchgeführt. Insgesamt nahmen 45 Kinder daran teil, davon 12 Mädchen und 33 Jungen. 24 Kinder waren neun Jahre alt, 16 Kinder zehn Jahre alt und fünf Kinder elf Jahre alt. Pro Kurs waren ca. vier Betreuende vor Ort, was einem angemessenen Betreuungsverhältnis von etwa 1:3 entspricht, sodass die Betreuenden ausreichend Zeit hatten, den Kindern weiterzuhelfen und die Fragen oder Schwierigkeiten während des Bauens bzw. Programmierens zu notieren. Neben den Schwierigkeiten notierten sie u.a. auch den Namen des Kindes, den Schritt, bei dem die Schwierigkeit auftrat, und die Dauer der Unterstützung. Jeweils nach dem Bauen (nach Einheit 2) bzw. nach dem Programmieren (nach Einheit 4) wird mithilfe des Fun Toolkit gemessen, wie viel Spass die Kinder hatten (Read 2008). Abbildung 5 zeigt die Bestandteile des Fun Toolkits.

Würdest du das Bauen im
Nachhinein wieder machen
wollen?

nein vielleicht ja

Wie hat dir das Bauen
gefallen?

				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
schlecht	nicht so gut	gut	sehr gut	perfekt

Abb. 5: Fun Toolkit zur Messung des Spasses beim Bauen.

5.2 Datenanalyse

Zur Beantwortung der Forschungsfragen werden die Notizen der Betreuenden und die von den Kindern ausgefüllten Fun Toolkits herangezogen. Die qualitativen Daten aus den Notizen der Betreuenden wurden mittels hermeneutischer Inhaltsanalyse in einem Mixed-Methods-Ansatz ausgewertet (Bergman 2010). Dazu wurde nach dem Lesen aller Notizen ein Kategoriensystem vereinbart und eine Forschende kategorisierte alle Notizen. 15 % der Notizen wurden von einem weiteren Forschenden kategorisiert. Es ergibt sich eine Interrater-Reliabilität von $K=0,64$. Die sich ergebende Item-Anzahl in den jeweiligen Kategorien wurde wiederum auf die ursprünglichen qualitativen Daten und die Forschungsfragen bezogen. Die Auswertung der Again Tables und der Smileyometer führte zu quantitativen Daten. Pro Abschnitt (à vier Stunden) und Kind ergibt sich ein Wert zwischen 0 und 2 bzw. 0 und 4. Die Korrelation der Spass-Werte mit der Anzahl der Schwierigkeiten wurde mittels Kendalls Tau-b mit einer Signifikanzschwelle von $\alpha=0,05$ berechnet, wobei $\tau >,1$ bzw. $<-,1$ schwache Effekte, $>,3$ bzw. $<-,3$ mittlere Effekte und $>,5$ bzw. $<-,5$ starke Effekte für die jeweilige Untergruppe anzeigt.

5.3 Einschränkungen

Der Kurs fand in einem ausserschulischen Rahmen statt, was die externe Validität der Untersuchung einschränkt. Es nahmen viele Kinder teil, die bereits andere naturwissenschaftlich-technische Kurse am Lernort besucht hatten. Neben teilweise vorhandenen Vorerfahrungen kann dies bedeuten, dass besonders interessierte oder stärker geförderte Kinder am Kurs teilnahmen, weshalb die Stichprobe nicht repräsentativ ist. Ausserdem beziehen sich die Schwierigkeiten der Kinder auf die Aktivitäten in unserem spezifischen Kurs und können je nach Schwerpunkt in anderen Robotik-Making-Kursen variieren. Um die konkreten Ergebnisse übertragbar zu machen, beziehen sich die abgeleiteten Empfehlungen bewusst auf grundlegende und übertragbare Ideen.

6. Empirische Befunde

6.1 RQ 1: Schwierigkeiten beim Bauen eines Roboters

Die Kategorisierung der Schwierigkeiten beim Bauen erfolgt anhand von drei Dimensionen: Um welche Tätigkeiten handelt es sich? Was ist die Ursache? Welche Hilfe wird von den Betreuenden geleistet?

Die mittlere Spalte der Tabellen 1–3 gibt an, auf wie viele Kinder die angegebene Kategorie ein- oder mehrmals zutrifft. Die letzte Spalte zeigt, wie oft diese Kategorie pro Kind aufgerufen wurde.

Art der Tätigkeit	Anteil der Kinder	Durchschnittliche Häufigkeit
Löten	75,6 %	2,4
Schrauben	73,3 %	1,9
Stecken	66,7 %	2,5
Leimen	51,1 %	1,6
Verkabeln	42,2 %	1,3
Schneiden	28,9 %	1,0
Drehen	6,7 %	1,0
Verbiegen	4,4 %	1,0

Tab. 1: Verteilung der Schwierigkeiten hinsichtlich der Tätigkeiten.

6.1.1 Tätigkeiten

Insgesamt baten 97,8 % der Kinder um Hilfe beim Bauen. Es zeigt sich, dass jeweils über die Hälfte der Kinder beim Löten, Schrauben, Stecken und Leimen Unterstützung benötigen. Dies mag an teilweise mangelnden Vorerfahrungen in diesen Techniken liegen, wobei aufgrund des außerschulischen und freiwilligen Rahmens eine grundsätzliche Affinität möglich wäre. Vor allem beim Löten und beim Stecken benötigen die Kinder häufiger Hilfe. Beim Stecken lässt sich die hohe Zahl dadurch erklären, dass in 20 von insgesamt 30 Schritten beim Bauen etwas gesteckt (also zusammengesteckt, hineingedrückt oder befestigt) wird. Gelötet wird dagegen nur in drei der 30 Schritte, jedoch besteht das Löten selbst aus mehreren Teilschritten. Diese können jeweils Anlass für Fragen wie «Wie soll ich das Kabel in die Abisolierzange reinschieben?» (S21) und Tipps wie «vorher verdrillen» (S16) und «die Enden vorher warm zu machen» (S12), «damit keine kalte Lötstelle entsteht» (S14) bieten. Die Vorerfahrungen der Kinder und die Mehrschrittigkeit von Tätigkeiten scheinen Einfluss auf die Häufigkeit von Schwierigkeiten bei einer Tätigkeit zu haben.

Art der Ursache	Anteil der Kinder	Durchschnittliche Häufigkeit
Hoher Schwierigkeitsgrad	33,3 %	1,4
Falsche Ausführung	26,7 %	1,2
Falsches Material	20,0 %	1,1
Mangelnde Vorerfahrung	6,7 %	1,0
Kursgeschwindigkeit	6,7 %	1,0
Ungenauere Ausführung	6,7 %	1,0

Tab. 2: Verteilung der Ursachen der Schwierigkeiten.

6.1.2 Ursachen

Für die Schwierigkeiten der Kinder wurden verschiedene Ursachen identifiziert. Diese lassen sich in hohen Schwierigkeitsgrad, falsche Ausführung, falsches bzw. defektes Material, mangelnde Vorerfahrung, Kursgeschwindigkeit und ungenaue Ausführung unterteilen (Tabelle 2). Es zeigt sich, dass jeweils ein Fünftel bis ein Drittel der Kinder Probleme aufgrund des hohen Schwierigkeitsgrades, der falschen Ausführung oder des falschen Materials hatte. Der hohe Schwierigkeitsgrad wiederum hat verschiedene Gründe, die materialimmanenter, physischer und kognitiver Natur sind: «Holzteile gingen schwer rein» (S19), «das Verbindungsstück musste mit Kraft reingedrückt werden» (S23) und «kurze Verwirrung, wo Sensor zu befestigen ist» (S4). Teilweise wurden zusätzlich auch explizite Ursachen angegeben wie die mangelnde Vorerfahrung oder die Kursgeschwindigkeit. Die falsche Ausführung bezieht sich beispielsweise auf die Positionierung: «Kabel auf die rechte Seite gesteckt

anstatt auf die linke» (S20) oder «unten [anstatt seitlich] geleimt» (S8). Ungenaue Ausführung dagegen meint, dass die Positionierung prinzipiell korrekt, allerdings nicht akkurat durchgeführt wurde, indem beispielsweise «Schrauben immer schief reingedreht» (S34) wurden. Das falsche bzw. defekte Material kann auf die versehentliche Bereitstellung von zu kurzen Schrauben, die versehentliche Auswahl falschen Materials durch die Kinder und auf von vornherein defekte Materialien zurückgeführt werden. Eine fehlende Anpassung des Schwierigkeitsgrads an die Merkmale der Kinder scheint problematisch zu sein.

6.1.3 Arten der Hilfe

Die Kinder benötigen verschiedene Arten von Hilfe durch die Betreuenden: Anleitung vor der eigentlichen Bearbeitung, Mithilfe während der Bearbeitung sowie Korrektur und Bestätigung nach der Bearbeitung (Tabelle 3). Darüber hinaus erkennen einzelne Kinder die Lösung direkt nach dem Stellen der Frage selbst und benötigen keine Hilfe mehr. Anleitung und Mithilfe wurden von den Betreuenden am häufigsten explizit notiert, für jeweils etwa die Hälfte der Kinder. Dass einige Kinder mehr Anleitung vor der eigentlichen Bearbeitung benötigten, kann damit erklärt werden, dass (1) die Kinder aufgrund ihrer teilweise geringen Vorerfahrungen unsicher waren und (2) im Kurs bewusst nicht zu viele Informationen vorab gegeben wurden, um die Problemlösung nicht vorwegzunehmen oder die Kinder zu unterfordern. Die Nachfragen bezogen sich auf die Art und Weise, die Reihenfolge, die Menge und den Ort, beispielsweise «Wie leimt man? Wie viel Leim [verwendet man]?» (S10). Bei der aktiven Mithilfe während der Bearbeitung wurde besonders häufig eine weitere Hand benötigt: «Beim Leimen musste man beim Halten helfen» (S42), «dritte Hand beim Löcher anschrauben» (S9) und «Kinder haben sich [gegenseitig] geholfen beim Leimen und Schrauben» (S30 und S32). Korrekturen kamen etwas seltener vor, was auch mit dem pädagogischen Konzept zusammenhängen kann, nicht ungefragt zu viel korrigierendes Feedback zu geben. Bestätigungen wurden nur auf Nachfrage gegeben. Besonders häufig wurde die Richtigkeit der Ausführung in Bezug auf die Qualität und den Ort korrigiert bzw. bestätigt. Allgemein scheinen die Kinder v.a. Hilfen zu brauchen, die wie die Anleitung auch durch einen schriftlichen Hinweis oder wie die Mithilfe auch durch Peers übernommen werden können. Für die Grundschule wurde bereits gezeigt, dass solche Hinweiskärtchen hilfreich sein können, aber auch Differenzierung erfordern (Greifenstein, Heuer, und Fraser 2024).

Art der Hilfe	Anteil der Kinder	Durchschnittliche Häufigkeit
Anleitung	53,3 %	1,8
Mithilfe	46,7 %	1,9
Korrektur	37,8 %	1,4
Bestätigung	20,0 %	1,3
Keine Hilfe mehr benötigt nach Stellen der Frage	4,4 %	1,5

Tab. 3: Verteilung der Arten der Hilfe.

6.2 RQ 2: Zusammenhang von Schwierigkeiten beim Bauen und Spass

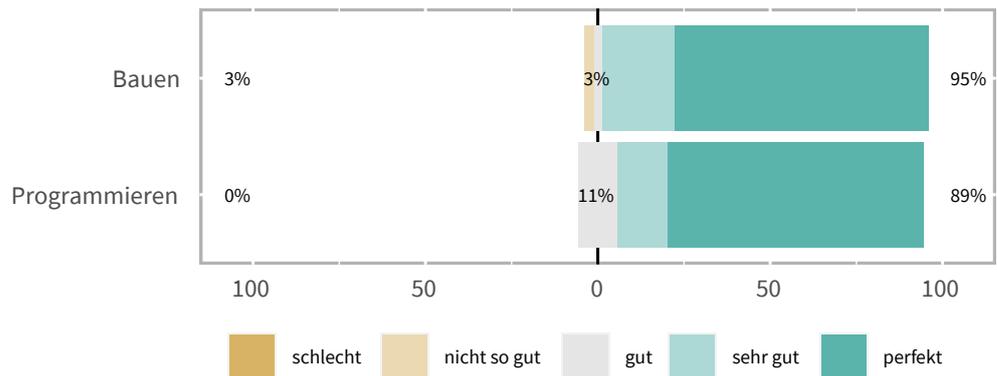


Abb. 6: Verteilung der Werte des Smileyometer für das Bauen und das Programmieren.

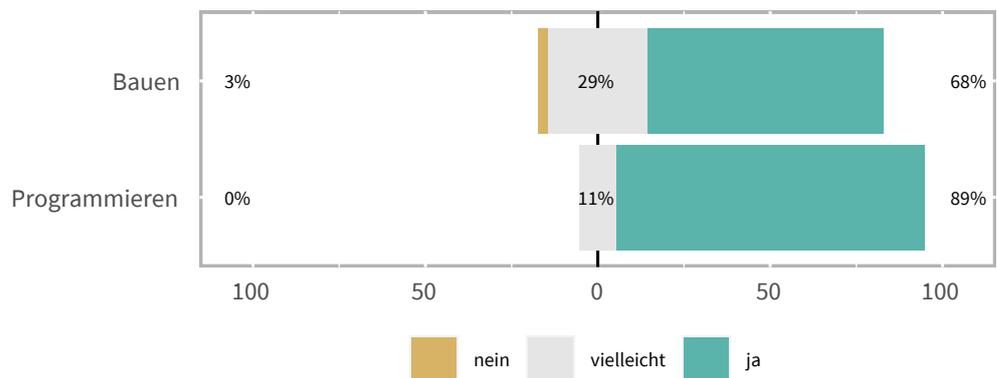


Abb. 7: Verteilung der Werte der Again Again Table für das Bauen und das Programmieren.

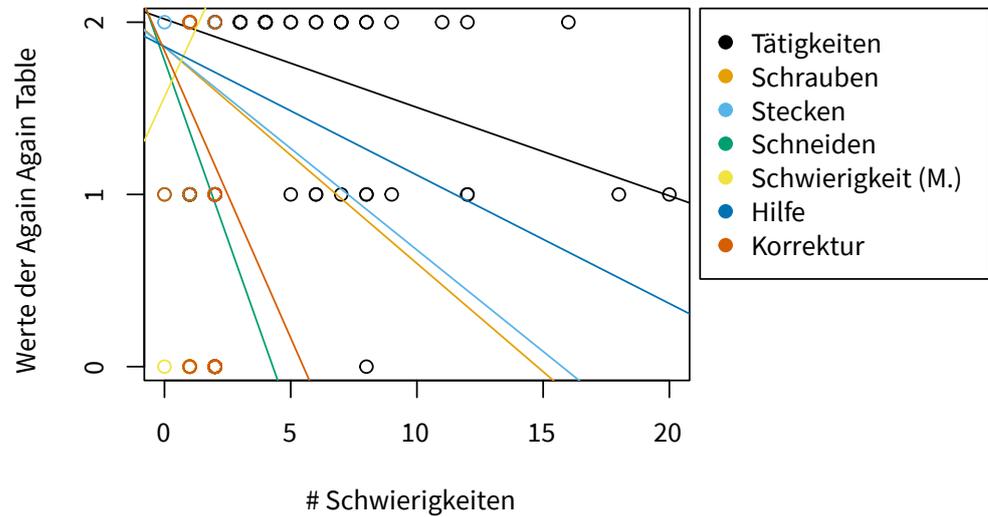


Abb. 8: Signifikante Korrelationen der Anzahl der Schwierigkeiten des Bauens mit den Werten der Again Again Table

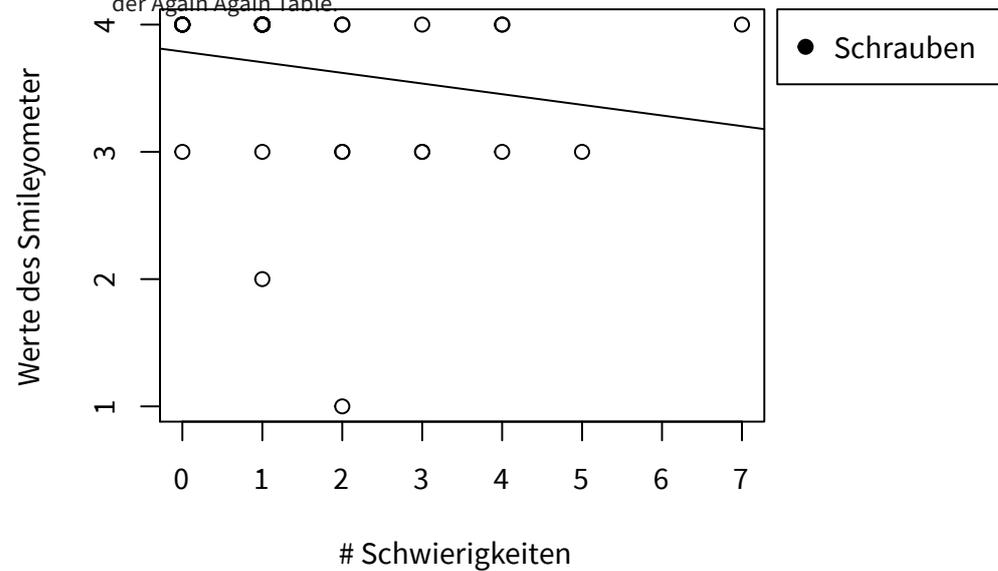


Abb. 9: Signifikante Korrelationen der Anzahl der Schwierigkeiten des Bauens mit den Werten des Smileyometer.

Zur Beantwortung von RQ 2 werden die Schwierigkeiten aus RQ 1 und die Einschätzung des Spasses durch die Kinder, gemessen mit dem «Smileyometer» und der «Again Again Table», herangezogen. Abbildung 6 und Abbildung 7 zeigen die Verteilung der Werte im Vergleich zum Bauen. Im Allgemeinen sind die Werte des Spasses, mit einem Mittelwert von $S = 3,66$ ($\bar{x} = 4$) für den Smileyometer und $A = 1,66$ ($\bar{x} = 2$) für die Again Again Table hoch. Abbildung 8 und Abbildung 9 zeigen alle Korrelationen, für die der zugehörige p -Wert signifikant ist.

Die Korrelation zwischen der Anzahl der Schwierigkeiten und dem Spass zeigt, dass eine höhere Anzahl mit einem signifikant schlechteren Wert in der Again Again Table zusammenhängt ($\tau = -,38$, $p = ,006$). Inwieweit sich dieses Ergebnis oder auch eine Korrelation mit dem Smileyometer in den verschiedenen Dimensionen der Schwierigkeiten beim Bauen wiederfindet, wird im Folgenden beschrieben und mögliche Erklärungen werden diskutiert.

6.2.1 Tätigkeiten

Die Spass-Werte der Again Again Table gehen bei steigender Anzahl an Schwierigkeiten bezüglich der Tätigkeiten insgesamt signifikant zurück. Dies gilt für die Tätigkeiten Stecken, Schrauben und Schneiden. Beim Schrauben sind die Effekte am grössten ($\tau = -,44$, $p = ,003$) und beim Schrauben sind auch die Smileyometer-Werte signifikant niedriger ($\tau = -,31$, $p = ,032$). Die Mittelwerte liegen bei $A = 1,57$ bzw. $S = 3,59$ mit Schwierigkeit(en) beim Schrauben und bei $A = 2,00$ bzw. $S = 3,89$ ohne Schwierigkeiten.

Schrauben und Stecken kommen am häufigsten in den Schritten des Zusammenbaus vor (Abbildung 1). Beides zusammen erfordert aber nur ähnlich viel Hilfestellung wie das Löten (Abbildung 1), das seltener in den Schritten vorkommt. Dadurch, dass nicht ein Grossteil der Gruppe, sondern jeweils nur einzelne Kinder während eines Schritts Unterstützung beim Schrauben und Stecken benötigen, könnte das Zusammengehörigkeitsgefühl und damit die soziale Eingebundenheit reduziert werden. Auch der soziale Vergleich kann dazu führen, dass sich die Kinder als weniger kompetent und weniger autonom wahrnehmen. Folglich kann vermutet werden, dass die drei Komponenten soziale Eingebundenheit, Kompetenzerleben und Autonomie, die zur Förderung intrinsischer Motivation notwendig sind (Deci und Ryan 1993), reduziert werden.

6.2.2 Ursachen

Die Fehlerursache *materialimmanenter Schwierigkeitsgrad* ist die einzige Kategorie, deren häufigere Nennung zu signifikant höheren Spass-Werten in der Again Again Table führt ($\tau = -,360$, $p = ,023$). Dies könnte daran liegen, dass der Fehler nicht dem eigenen Unvermögen, sondern dem Material zugeschrieben wird. Zudem wird den Kindern vor Augen geführt, dass auch Erwachsene Schwierigkeiten haben, wenn Holzteile beispielsweise nicht perfekt in eine Aussparung passen. Das wiederum mindert nicht das Kompetenzerleben, sondern könnte im Gegenteil auch in anderen Situationen des vermeintlichen Misserfolgs zu einer Verschiebung der Attribuierung von internalen hin zu externalen Ursachen (Weiner 1985) führen.

6.2.3 Arten der Hilfe

Die Anzahl der Hilfestellungen korreliert mit den Spass-Werten der Again Again Table. Letztere sind signifikant niedriger im Fall einer hohen Anzahl von Hilfestellungen im Gesamten sowie der Korrektur im Speziellen. Die Unterschiede bezüglich der Korrektur ($\tau = -,450$, $p = ,004$) zeigen sich auch in dem vergleichsweise niedrigen Mittelwert von $A = 1,36$ bei Kindern, die korrigierendes Feedback bekommen haben und dem Mittelwert von $A = 1,83$ bei Kindern ohne korrigierendes Feedback. Das zeigt auf, dass Kinder, die korrigiert wurden, weniger Spass empfinden. Auch wenn die Korrektur nicht die Ursache sein muss, so macht diese Art von Feedback Kindern jedoch bewusst, dass sie auf Hilfe angewiesen sind, was ihre wahrgenommene Autonomie reduzieren kann. Zum anderen zeigt eine Korrektur, dass etwas falsch gemacht wurde, was dazu führen kann, dass das Kind seine Kompetenzen infrage stellt. Beide Komponenten, Autonomie und Kompetenzerleben, könnten folglich reduziert sein.

6.3 RQ 3: Schwierigkeiten beim Programmieren eines Roboters

Die Kategorisierung der Schwierigkeiten beim Programmieren erfolgt anhand von drei Dimensionen: Um welches Programmierkonzept bzw. welchen Teilbereich der Programmierumgebung handelt es sich und welche Hilfe wird von den Betreuenden geleistet? Tabelle 4 gibt hierzu einen Überblick.

Programmierkonzepte	Anteil der Kinder	Durchschnittliche Häufigkeit
Roboterspezifische Konzepte	84,4 %	3,7
• Bewegung	66,7 %	3,1
• Beleuchtung	48,9 %	1,6
• Sensoren	28,9 %	1,1
Algorithmische Konzepte	66,7 %	1,8
• Verzweigung	35,6 %	1,2
• Wiederholung	24,4 %	1,3
• Warten	22,2 %	1
• Reihenfolge	15,6 %	1
• Sonstiges	6,7 %	1
Allgemein	20,0 %	1,4
Nachbauen	13,3 %	1,2

Tab. 4: Verteilung der Schwierigkeiten hinsichtlich der Konzepte.

6.3.1 Programmierkonzepte

93,3 % der Kinder bitten um Hilfe bei roboterspezifischen Konzepten und/oder algorithmischen Konzepten. Die Unterkategorien zeigen, dass deren Häufigkeit nicht nur von der Häufigkeit des Auftretens in den Aufgaben abhängt (Abbildung 3). So ist die Beleuchtung eine relativ häufige Schwierigkeit, zu den Sensoren haben die Kinder relativ selten Fragen. Dies lässt sich dadurch erklären, dass (1) die Beleuchtung in der ersten Programmieraufgabe des Kurses vorkommt und (2) in der zweiten Aufgabe angepasst werden muss. Im Gegensatz dazu werden die Sensoren nur innerhalb der Use-Aufgaben verwendet und müssen folglich zwar nachgebaut, aber nicht verändert werden. Auch bei den algorithmischen Konzepten sind die Häufigkeiten anders als aufgrund der Vorkommenshäufigkeiten in den Aufgaben (Abbildung 3) zu erwarten wäre. So benötigen mehr Kinder Unterstützung bei der Verzweigung wie «wohin [füge ich] fahr-Blöcke in den wenn-dann-sonst-Block [ein]?» als bei der Wiederholung, obwohl letztere häufiger vorkommt (Abbildung 3). Zwar haben Lernende sowohl häufig Schwierigkeiten mit Wiederholungen als auch mit Verzweigungen (Fraser 2015; Grover und Basu 2017), allerdings enthält die Verzweigung in den Aufgaben zudem Variablen und Boolesche Operatoren, die ebenfalls Schwierigkeiten bereiten können (Grover und Basu 2017). Die Endloswiederholung in den Aufgaben dagegen benötigt keine weiteren Blöcke. Insgesamt scheint die Komplexität der Inhalte einen starken Einfluss auf die Häufigkeit von Schwierigkeiten beim Programmieren zu haben.

Schwierigkeiten bezüglich Programmierumgebung	Anteil der Kinder	Durchschnittliche Häufigkeit
Finden von Blöcken	24,4 %	1,1
Zusammenfügen	17,8 %	1
Verbindung	11,1 %	1,4
Allgemeine Bedienung	8,9 %	1,2
Eigenheiten der Programmierumgebung	8,9 %	1
Sonstiges	6,6 %	1

Tab. 5: Verteilung der Schwierigkeiten bezüglich der Programmierumgebung.

6.3.2 Programmierumgebung

Die Programmierumgebung mBlock wird für alle Aufgaben benötigt und 46,7% der Kinder haben mindestens einmal eine Frage dazu. Dabei traten verschiedene Schwierigkeiten auf wie «findet grüne Blöcke nicht» (S34), «versteht nicht, wie man Block in Lücke schiebt» (S10), «Hochladen funktioniert nicht» (S33), «mBlock-Umgebung

erklärt» (S3) und kleinere Eigenheiten von mBlock wie «warte-Block Sekunde(n) auf 0.5 setzen (Punkt statt Komma)» (S25). Dies kann damit zusammenhängen, dass die technische Medienkompetenz erst ab einem Alter von zehn Jahren entsprechend ausgeprägt ist (Feierabend et al. 2021) und die Kinder daher noch mehr Hilfestellungen benötigen.

Art der Hilfe	Anteil der Kinder	Durchschnittliche Häufigkeit
Korrektur	71,1 %	1,4
Anleitung	46,7 %	2
Umgang mit Hardware-Problemen	40,0 %	1,5
Mithilfe	20,0 %	1,4
Bestätigung	8,9 %	1

Tab. 6: Verteilung der Arten der Hilfe.

6.3.3 Arten der Hilfe

Ähnlich wie beim Zusammenbauen, benötigen die Kinder beim Programmieren verschiedene Arten von Hilfen durch die Betreuenden: Korrektur, Anleitung, Mithilfe und Bestätigung, aber auch Umgang mit Hardware-Problemen wie «Kabel falsch eingesteckt» (S2), «Ultraschallsensor defekt (S44) oder «Kabel runtergefliegen – wieder rangelötet» (S16). Auffallend ist, dass mehr als zwei Drittel der Kinder Korrektur einfordern bzw. benötigen. Das könnte damit zu erklären sein, dass durch das Testen mit dem Roboter direkt ersichtlich wird, dass etwas falsch gemacht wurde. Fehlerursachen waren falsche Werte (z.B. «Werte für richtige Geschwindigkeit» (S15)) und fehlerhafte Programmiermuster wie *LED Off Missing* (Obermüller et al. 2022), bei dem der zugehörige Pin einer LED «zwei Mal auf hoch gesetzt» (S44) wird, sie damit «an und an» (S18) gemacht wird, und die LED deswegen nicht blinkt. Zwar benötigen weniger Kinder eine Anleitung als eine Korrektur, aber wenn ein Kind eine Anleitung benötigt, dann durchschnittlich zweimal (Tabelle 6). Das könnte darauf zurückgeführt werden, dass unerfahrenere Kinder vor der Bearbeitung neben einer genauen Erklärung der Programmierumgebung auch zu den Programmierkonzepten eingehende Informationen benötigen. Die Bewusstheit der Kinder, dass sie Anleitung bzw. Korrektur benötigen, scheint folglich Einfluss auf die Häufigkeit ihrer Fragen zu haben.

6.4 RQ 4: Zusammenhang von Schwierigkeiten beim Programmieren und Spass

Zur Beantwortung von RQ 4 werden die Schwierigkeiten aus RQ 3 und die Einschätzung des Spasses durch die Kinder, gemessen mit dem «Smileyometer» und der «Again Again Table», herangezogen. Im Allgemeinen sind die Spasswerte, mit einem Mittelwert von 3,63 ($\bar{x}=4$) nach dem Smileyometer und 1,89 ($\bar{x}=2$) nach der Again Again Table hoch. Abbildung 8 und Abbildung 9 zeigen alle Korrelationen, deren p -Wert signifikant ist. Diese werden im Folgenden erläutert.

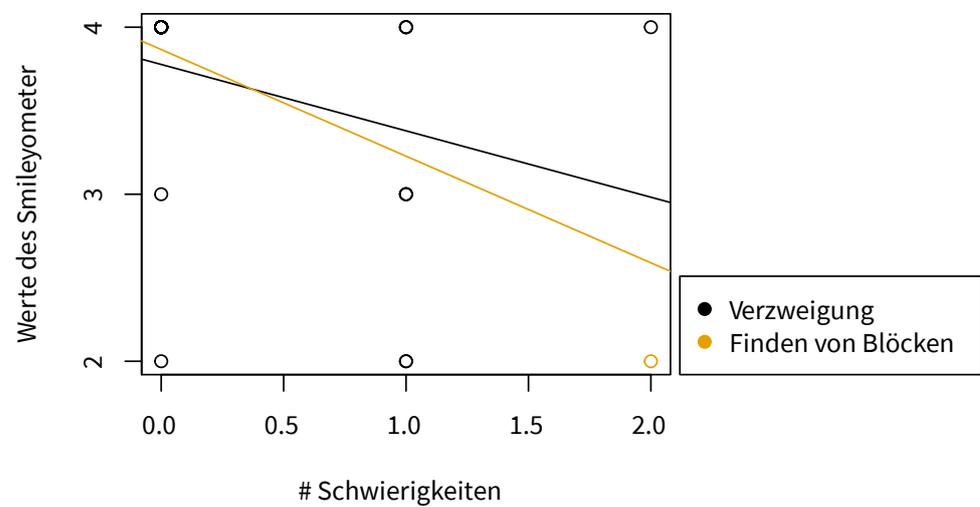


Abb. 10: Signifikante Korrelationen der Anzahl der Schwierigkeiten des Programmierens mit den Werten des Smileyometer.

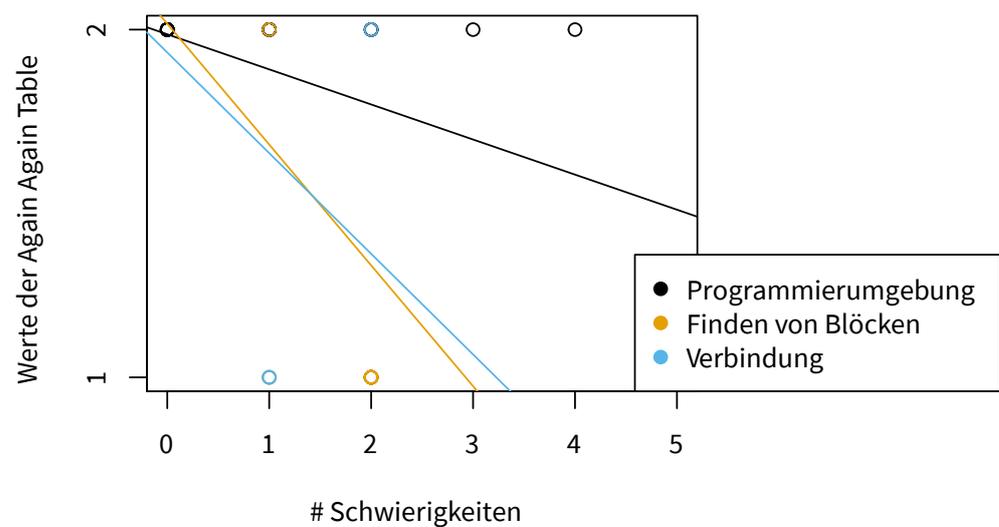


Abb. 11: Signifikante Korrelationen der Anzahl der Schwierigkeiten des Programmierens mit den Werten der Again Again Table.

6.4.1 Programmierkonzepte

Bei den Programmierkonzepten korreliert nur die steigende Anzahl an Schwierigkeiten bezüglich der Verzweigung mit signifikant niedrigeren Werten des Smileyometer ($\tau = -,41$, $p = ,028$). Dies kann daran liegen, dass das Konzept einer Verzweigung, wie beschrieben, relativ komplex ist und somit das Kompetenzerleben erschwert.

6.4.2 Programmierumgebung

Eine steigende Anzahl der Schwierigkeiten bezüglich der Programmierumgebung insgesamt, des Findens von Blöcken und der Verbindung sind mit signifikant schlechteren Werten der Again Again Table verbunden. Die Effekte sind dabei für die Verbindung am höchsten ($\tau = -,59$, $p = ,002$) und das Finden von Blöcken korreliert zusätzlich mit schlechteren Werten des Smileyometers ($\tau = -,37$, $p = ,046$). Insbesondere Probleme bei der Verbindung lassen sich nicht auf die eigenen Fähigkeiten zurückführen, weshalb das Kompetenzerleben hierdurch nicht eingeschränkt wird. Schwierigkeiten mit der Programmierumgebung können jedoch generell zu einer Einschränkung der Autonomie führen, da die Bedienung der Programmierumgebung eine grundlegende Voraussetzung für das Programmieren in dieser ist. Dringende Schwierigkeiten könnten folglich den Spass reduzieren, was auch für andere Roboter bestätigt wurde (Greifenstein et al. 2022).

6.4.3 Art der Hilfe

Im Vergleich zum Zusammenbauen scheint die Art der Hilfe beim Programmieren weniger wichtig für den erfahrenen Spass zu sein. Eine mögliche Erklärung dafür ist, dass die Betreuenden beim Programmieren pro Schwierigkeit durchschnittlich eine Minute und 45 Sekunden ($\bar{x} = 60$) verbrachten, beim Zusammenbauen hingegen nur 37 Sekunden ($\bar{x} = 15$). Da die Kinder beim Programmieren relativ viel Unterstützung durch die Betreuenden erhielten, legt der soziale Vergleich nahe, dass die Schwierigkeiten nicht an den individuellen Fähigkeiten liegen, weshalb das Kompetenzerleben wie auch die soziale Eingebundenheit weniger stark reduziert sein könnten. Es bleibt zu untersuchen, inwieweit hier das Setting im außerschulischen Rahmen oder der Gruppendynamik im Klassenverband eine Rolle spielt. Eine weitere Erklärung dafür, dass Korrekturen nicht mit schlechteren Spasswerten korrelieren, könnte sein, dass die Kinder ihre Fehler durch das Fehlverhalten des Roboters selbst bemerkten und nicht von den Betreuenden darauf hingewiesen werden mussten. Stattdessen erhielten sie von den Betreuenden nur informatives Feedback, das wiederum das Kompetenzerleben weniger schwächt als ein (negativ) bewertendes Feedback.

7. Diskussion von Handlungsempfehlungen

Ausgehend von den empirischen Befunden dieser Studie und verwandten Ergebnissen werden im Folgenden Handlungsempfehlungen für die Unterstützung von Kindern diskutiert.

7.1 *Der Schwierigkeitsgrad sollte angemessen sein*

Da der Schwierigkeitsgrad der Aufgabenstellung Einfluss auf die Häufigkeit von Schwierigkeiten hat, sollte insbesondere bei mehrschrittigen Tätigkeiten und komplexen Inhalten auf einen angemessenen Schwierigkeitsgrad geachtet werden, damit sich die Lernenden als kompetent erleben. Im Making-Kontext erfolgt diese Anpassung vor der Bearbeitung, wenn die Lernenden beispielsweise feststellen, dass ihr geplantes Muster zu komplex ist (Spieler et al. 2020), oder während der Bearbeitung, wenn sich unlösbare Schwierigkeiten auftun (Blikstein 2013). Besonders Letzteres ist sehr zeitintensiv und daher in der Schule aufgrund der curricularen Vorgaben und der 45- bis 90-Minuten-Taktung üblicherweise schwer umsetzbar ist (Ingold und Maurer 2019). Ein Mittelweg könnte der Einsatz von vorstrukturierten Aufgaben sein, die die Merkmale der Maker-Education – selbstgesteuert, spielerisch und handlungsorientiert – (Ingold und Maurer 2019) erfüllen. Dies könnte erreicht werden, indem die Aufgaben zwar vorgegeben werden, aber beispielsweise Freiheiten in der Ausgestaltung und der Problemlösung oder bei Bedarf Hinweise zur Unterstützung gegeben werden.

7.2 *Die Vorerfahrungen der Lernenden sollten berücksichtigt werden*

Die (zusätzliche) Unterstützung sollte nach den individuellen Kompetenzen der Lernenden differenziert werden. Gerade bei einer für viele Kinder neuen Thematik wie dem Physical Computing wird der Differenzierungsbedarf allerdings oft erst im Verlauf des Kurses deutlich. Daher sollten allgemeine Erkenntnisse zu typischen Problemen beim blockbasierten Programmieren (Frädrich et al. 2020; Obermüller et al. 2022) oder das allgemeine Vorwissen zum Programmieren (Geldreich, Simon, und Starke 2019) und zur Medienkompetenz (Feierabend et al. 2021) in die zusätzliche Unterstützung wie vorbereitete Hinweise einfließen. Durch vorbereitete Hinweise können Lehrkräfte zudem entlastet werden, denn die Unterstützung der Lernenden während des Unterrichts wird häufig als eine der Herausforderungen des Programmierunterrichts gesehen (Michaeli und Romeike 2019; Greifenstein, Graßl, und Fraser 2021; Sentance und Csizmadia 2017; Yadav et al. 2016).

7.3 *Es sollte eine positive Fehlerkultur vorherrschen*

Da korrigierendes Feedback für den Kompetenzerwerb wichtig ist (Wisniewski, Zierer, und Hattie 2020), aber auch den erlebten Spass beeinflussen kann, sollte über den Umgang mit Fehlern nachgedacht werden. Damit Fehler weniger als negativ, sondern als wichtiger Bestandteil des Lernprozesses wahrgenommen werden, können interessante Fehler von verschiedenen Schüler:innen im Unterrichtsgespräch besprochen oder an einer Pinnwand gewürdigt werden; auch sollten Fehler der Lehrkraft thematisiert werden (Vivian, Grover, und Falkner 2020). Durch den sozialen Vergleich, dass Fehler auch anderen kompetenten Personen passieren können, wird den Schüler:innen gezeigt, dass Fehler nicht im Gegensatz zu Kompetenzerleben stehen müssen. Um die negative Konnotation von Fehlern weiter abzuschwächen, sollte das Feedback konstruktiv und wertschätzend vermittelt werden. So könnte automatisiertes Feedback (wie durch das Verhalten des Roboters in dieser Studie oder verschiedene Formen computergestützten Feedbacks) das Kompetenzerleben weniger reduzieren, da es im Vergleich zum menschlichen Feedback als weniger bedrohlich wahrgenommen wird (Blok et al. 2002). Dies wiederum könnte Lehrkräften Entlastung bei der Unterstützung ihrer Schüler:innen bieten: Sie kennen zwar tendenziell wenige automatisierte Werkzeuge, schätzen diese aber grundsätzlich als sehr hilfreich ein (Greifenstein, Graßl, und Fraser 2021).

7.4 *Massnahmen zur Unterstützung sollten die Lernenden kognitiv aktivieren*

Sowohl bei automatisiertem als auch bei nicht automatisiertem Feedback sollte auf eine kognitive Aktivierung und dadurch auf ein gewisses Mass an Autonomie der Lernenden geachtet werden. Dies kann beispielsweise durch Aufforderungen geschehen, den Hinweis mit eigenen Worten zu erklären (Marwan, Williams, und Price 2019), durch Hinweise zur Vorgehensweise statt direkter Handlungsanweisungen (Greifenstein et al. 2022) oder durch zeitversetztes Feedback, sodass Lernende sich nicht ausschliesslich darauf konzentrieren (Chevalier et al. 2022). Die zeitliche Verzögerung gilt jedoch nicht für dringende Probleme wie ein grundlegendes Unverständnis bezüglich der Aufgabenstellung oder der Programmierumgebung. Hier muss darauf geachtet werden, dass diese zeitnah behoben werden, da sie ansonsten die Autonomie und damit zusammenhängend den Spass stark einschränken können (Greifenstein et al. 2022).

8. Fazit und Ausblick

In diesem Beitrag wurde ein Making-Kurs und dessen Evaluation im Hinblick auf die Förderung affektiver Faktoren beschrieben. Es zeigte sich, dass beim Zusammenbauen eines Roboters und dessen Programmierung verschiedene Schwierigkeiten

auftreten, deren Häufigkeit von Kriterien wie dem Schwierigkeitsgrad, den Vorerfahrungen und dem pädagogischen Unterstützungskonzept abhängt. Während Schwierigkeiten einerseits zum Problemlöseprozess gehören, können bestimmte Schwierigkeiten andererseits den Spass der Lernenden reduzieren, dessen Förderung neben anderen affektiven Komponenten beim Physical Computing in der Grundschule besonders wichtig ist. Deswegen sollten insbesondere Kompetenzerleben und Autonomie ermöglicht werden, um positive Emotionen wie Spass hervorzurufen, die sich längerfristig schliesslich vom situationalen Interesse zu individuellem Interesse und intrinsischer Motivation entwickeln. Für die Umsetzung eines affektiv förderlichen Unterrichts wurden unter Berücksichtigung der empirischen Befunde dieser Studie und verwandter Arbeiten verschiedene Handlungsempfehlungen diskutiert. Während einzelne Empfehlungen für die Vermittlung von Feedback bereits vorliegen, könnten weitere Forschungsarbeiten explizit die Perspektive der Lernenden betrachten, um diese in die Entwicklung konstruktiven Feedbacks einzubeziehen.

Literatur

- Amo, Daniel, Paul Fox, David Fonseca, und César Poyatos. 2020. «Systematic review on which analytics and learning methodologies are applied in primary and secondary education in the learning of robotics sensors». *Sensors* 21 (1): 153. <https://doi.org/10.3390/s21010153>.
- Aroca, Rafael V, Rafael B Gomes, Dalton M Tavares, Anderson Abner S Souza, Aquiles MF Burlamaqui, Glauco AP Caurin, und Luiz MG Goncalves. 2012. «Increasing students' interest with low-cost cellbots». *IEEE Transactions on Education* 56 (1): 3–8. <https://doi.org/10.1109/TE.2012.2214782>.
- Bergman, Max. 2010. «Hermeneutic Content Analysis: Textual and Audiovisual Analyses Within a Mixed Method Framework». *Sage Handbook of Mixed Methods in Social & Behavioral Research. Second edition. Sage Publication*.
- Best, Alexander, Christian Borowski, Katrin Büttner, Rita Freudenberg, Martin Fricke, Kathrin Haselmeier, Henry Herper, u. a. 2019. «Kompetenzen für informatische Bildung im Primarbereich». [Report], Bonn: Gesellschaft für Informatik e.V.
- Blikstein, Paulo. 2013. «Digital fabrication and 'making' in education: The democratization of invention». *FabLabs: Of machines, makers and inventors* 4 (1): 1–21. <https://doi.org/10.14361/transcript.9783839423820.203>.
- Blok, Henk, Ron Oostdam, Martha E Otter, und Marianne Overmaat. 2002. «Computer-assisted instruction in support of beginning reading instruction: A review». *Review of educational research* 72 (1): 101–30. <https://doi.org/10.3102/00346543072001101>.
- Bunke-Emden, Hannah. 2020. «Potenziale von Making-Aktivitäten in informellen Lernumgebungen für die Medienpädagogik: Ergebnisse einer qualitativen Studie im Rahmen der Maker Days for Kids Leipzig». *Medienimpulse* 58 (4): 23-Seiten. <https://doi.org/10.21243/mi-04-20-11>.

- Catlin, Dave, Martin Kandhofer, Stephanie Holmquist, Andrew Paul Csizmadia, Julian Angel-Fernandez, und J Cabibihan. 2018. «Edurobot taxonomy and Papert's paradigm». *Constructionism* 2018: 151–59.
- Chalmers, Christina. 2018. «Robotics and computational thinking in primary school». *International Journal of Child-Computer Interaction* 17: 93–100. <https://doi.org/10.1016/j.ijcci.2018.06.005>.
- Chevalier, Morgane, Christian Giang, Laila El-Hamamsy, Evgeniia Bonnet, Vaios Papaspyros, Jean-Philippe Pellet, Catherine Audrin, Margarida Romero, Bernard Baumberger, und Francesco Mondada. 2022. «The role of feedback and guidance as intervention methods to foster computational thinking in educational robotics learning activities for primary school». *Computers & Education* 180: 104431. <https://doi.org/10.1016/j.compedu.2022.104431>.
- Deci, Edward L, und Richard M Ryan. 1993. «Die Selbstbestimmungstheorie der Motivation und ihre Bedeutung für die Pädagogik». *Zeitschrift für Pädagogik* 39 (2): 223–38. <https://doi.org/10.25656/01:11173>.
- Döbeli Honegger, Beat, und Michael Hielscher. 2017. «Vom Lehrplan zur Lehrerinnenbildung – erste Erfahrungen mit obligatorischer Informatikdidaktik für angehende Schweizer Primarlehrerinnen». *Informatische Bildung zum Verstehen und Gestalten der digitalen Welt*. 97-107, Gesellschaft für Informatik, Bonn.
- Du, Yuemeng, Andrew Luxton-Reilly, und Paul Denny. 2020. «A review of research on Parsons problems». In *Proceedings of the twenty-second Australasian Computing Education Conference*, 195–202. <https://doi.org/10.1145/3373165.3373187>.
- Feierabend, Sabine, T Rathgeb, H Kheredmand, und S Glöckler. 2021. «KIM-Studie 2020. Kindheit, Internet, Medien. Basisuntersuchung zum Medienumgang 6-bis 13-Jähriger in Deutschland». Medienpädagogischer Forschungsverbund Südwest (mpfs). Landesanstalt für Kommunikation Baden-Württemberg (LFK), Stuttgart.
- Frädrich, Christoph, Florian Obermüller, Nina Körber, Ute Heuer, und Gordon Fraser. 2020. «Common bugs in scratch programs». In *Proceedings of the 2020 ACM conference on innovation and technology in computer science education*, 89–95. <https://doi.org/10.1145/3341525.3387389>.
- Fraser, Neil. 2015. «Ten things we've learned from Blockly». In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, 49–50. IEEE. <https://doi.org/10.1109/BLOCKS.2015.7369000>.
- Geldreich, Katharina, Alexandra Simon, und Elena Starke. 2019. «Which Perceptions Do Primary School Children Have about Programming?». In *Proceedings of the 14th Workshop in Primary and Secondary Computing Education*, 1–7. <https://doi.org/10.1145/3361721.3361728>.
- Greifenstein, Luisa, Isabella Graßl, und Gordon Fraser. 2021. «Challenging but Full of Opportunities: Teachers' Perspectives on Programming in Primary Schools». In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research*, 1–10. <https://doi.org/10.1145/3488042.3488048>.

- Greifenstein, Luisa, Isabella Graßl, Ute Heuer, und Gordon Fraser. 2022. «Common Problems and Effects of Feedback on Fun When Programming Ozobots in Primary School». In *Proceedings of the 17th Workshop in Primary and Secondary Computing Education*, 1–10. <https://doi.org/10.1145/3556787.3556860>.
- Greifenstein, Luisa, Isabella Grassl, Ute Heuer, und Gordon Fraser. 2024. «“Help Me Solve It” or “Solve It For Me”: Effects of Feedback on Children Building and Programming Robots». In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 1–10. <https://doi.org/10.1145/3626252.3630752>.
- Greifenstein, Luisa, Ute Heuer, und Gordon Fraser. 2024. «Hint Cards for Common Ozobot Robot Issues: Supporting Feedback for Learning Programming in Elementary Schools.» In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 1–10. <https://doi.org/10.1145/3626252.3630868>.
- Grover, Shuchi, und Satabdi Basu. 2017. «Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic». In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*, 267–72. <https://doi.org/10.1145/3017680.3017723>.
- Hainey, Thomas, Thomas M Connolly, Elizabeth A Boyle, Amanda Wilson, und Aisya Razak. 2016. «A systematic literature review of games-based learning empirical evidence in primary education». *Computers & Education* 102: 202–23. <https://doi.org/10.1016/j.compedu.2016.09.001>.
- Heintz, Fredrik, Linda Mannila, und Tommy Färnqvist. 2016. «A review of models for introducing computational thinking, computer science and computing in K-12 education». In *FIE '16*, 1–9. <https://doi.org/10.1109/FIE.2016.7757410>.
- Ingold, Selina, und Björn Maurer. 2019. «Making in der Schule. Reibungspunkte und Synergieeffekte». *Chance Makerspace—Making trifft auf Schule*. 59–86. München: kopaed.
- Knaus, Thomas, und Jennifer Schmidt. 2020. «Medienpädagogisches Making: ein Begründungsversuch». *Medienimpulse* 58 (4): 50 Seiten. <https://doi.org/10.21243/mi-04-20-04>.
- Köller, Olaf, Felicitas Thiel, Isabell van Ackeren, Yvonne Anders, Michael Becker-Mrotzek, Ulrike Cress, Claudia Diehl, u. a. 2022. «Digitalisierung im Bildungssystem: Handlungsempfehlungen von der Kita bis zur Hochschule. Gutachten der Ständigen Wissenschaftlichen Kommission der Kultusministerkonferenz (SWK)». 185 pages. <https://doi.org/10.25656/01:25273>.
- Lee, Irene, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, und Linda Werner. 2011. «Computational thinking for youth in practice». *Acm Inroads* 2 (1): 32–37. <https://doi.org/10.1145/1929887.1929902>.
- Levy, Ronit Ben-Bassat, und Mordechai Ben-Ari. 2015. «Robotics activities—Is the investment worthwhile?» In *Informatics in Schools. Curricula, Competences, and Competitions: 8th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2015, Ljubljana, Slovenia, September 28-October 1, 2015, Proceedings* 8, 22–31. Springer. https://doi.org/10.1007/978-3-319-25396-1_3.

- Long, Ju. 2007. «Just For Fun: using programming games in software programming training and education». *Journal of Information Technology Education: Research* 6 (1): 279–90. <https://doi.org/10.28945/216>.
- Marwan, Samiha, Joseph Jay Williams, und Thomas Price. 2019. «An evaluation of the impact of automated programming hints on performance and learning». In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, 61–70. <https://doi.org/10.1145/3291279.3339420>.
- Michaeli, Tilman, und Ralf Romeike. 2019. «Current status and perspectives of debugging in the k12 classroom: A qualitative study». In *2019 IEEE Global Engineering Education Conference (Educon)*, 1030–38. IEEE. <https://doi.org/10.1109/EDUCON.2019.8725282>.
- Nenner, Christin, und Nadine Bergner. 2022. «Informatics Education in German Primary School Curricula». In *Informatics in Schools. A Step Beyond Digital Education: 15th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2022, Vienna, Austria, September 26–28, 2022, Proceedings*, 3–14. Springer. https://doi.org/10.1007/978-3-031-15851-3_1.
- Obermüller, Florian, Robert Pernerstorfer, Lisa Bailey, Ute Heuer, und Gordon Fraser. 2022. «Common Patterns in Block-Based Robot Programs». In *Proceedings of the 17th Workshop in Primary and Secondary Computing Education*, 1–10. <https://doi.org/10.1145/3556787.3556859>.
- Przybylla, Mareen, und Ralf Romeike. 2014. «Physical Computing and Its Scope—Towards a Constructionist Computer Science Curriculum with Physical Computing.» *Informatics in Education* 13 (2): 241–54. <https://doi.org/10.15388/infedu.2014.05>.
- Read, Janet C. 2008. «Validating the Fun Toolkit: an instrument for measuring children’s opinions of technology». *Cognition, Technology & Work* 10: 119–28. <https://doi.org/10.1007/s10111-007-0069-9>.
- Renninger, K Ann. 2009. «Interest and identity development in instruction: An inductive model». *Educational psychologist* 44 (2): 105–18. <https://doi.org/10.1080/00461520902832392>.
- Riedo, Fanny, Morgane Chevalier, Stéphane Magnenat, und Francesco Mondada. 2013. «Thymio II, a robot that grows wiser with children». In *2013 IEEE workshop on advanced robotics and its social impacts*, 187–93. IEEE. <https://doi.org/10.1109/ARSO.2013.6705527>.
- Scaradozzi, David, Laura Sorbi, Anna Pedale, Mariantonietta Valzano, und Cinzia Vergine. 2015. «Teaching robotics at the primary school: an innovative approach». *Procedia-Social and Behavioral Sciences* 174: 3838–46. <https://doi.org/10.1016/j.sbspro.2015.01.1122>.
- Selby, Cynthia, und John Woollard. 2014. «Refining an understanding of computational thinking». [E-Print] 1-23.
- Sentance, Sue, und Andrew Csizmadia. 2017. «Computing in the curriculum: Challenges and strategies from a teacher’s perspective». *Education and Information Technologies* 22: 469–95. <https://doi.org/10.1007/s10639-016-9482-0>.

- Spieler, Bernadette, Vesna Krnjic, Wolfgang Slany, Karin Horneck, und Ute Neudorfer. 2020. «Design, Code, Stitch, Wear, and Show It! Mobile Visual Pattern Design in School Contexts». In *2020 IEEE Frontiers in Education Conference (FIE)*, 1–9. IEEE. <https://doi.org/10.1109/FIE44824.2020.9274120>.
- Spieler, Bernadette, Tobias M Schifferle, und Manuela Dahinden. 2022. «Exploring Making in Schools: A Maker-Framework for Teachers in K12». In *6th FabLearn Europe/MakeEd Conference 2022*, 1–6. <https://doi.org/10.1145/3535227.3535234>.
- Vivian, Rebecca, Shuchi Grover, und Katrina Falkner. 2020. «Knowledge, Skills, Attitudes, & Beliefs: Learning Goals for Introductory Programming». In *Computer Science in K-12: An A to Z Handbook on Teaching Programming*, 113–24. Edfinity.
- Weiner, Bernard. 1985. «An attributional theory of achievement motivation and emotion.» *Psychological review* 92 (4): 548. <https://doi.org/10.1037/0033-295X.92.4.548>.
- Wisniewski, Benedikt, Klaus Zierer, und John Hattie. 2020. «The power of feedback revisited: A meta-analysis of educational feedback research.» *Frontiers in Psychology* 10: 3087. <https://doi.org/10.3389/fpsyg.2019.03087>.
- Yadav, Aman, Sarah Gretter, Susanne Hambrusch, und Phil Sands. 2016. «Expanding computer science education in schools: understanding teacher experiences and challenges.» *Computer science education* 26 (4): 235–54. <https://doi.org/10.1080/08993408.2016.1257418>.