

Informatikunterricht in der Grundschule? – Erprobung und Auswertung eines Unterrichtsmoduls mit Calliope mini

Kevin Baum¹, Nadine Kirsch¹, Kerstin Reese¹, Pascal Schmidt¹, Lukas Wachter¹, Verena Wolf¹

Abstract: Im Zuge der Digitalisierung wird die Lebenswelt von Grundschulkindern mehr und mehr durch Informatiksysteme geprägt. Informatische Bildung in der Grundschule kann zur Entmystifizierung solcher Systeme maßgeblich beitragen und so die Grundlage für tiefgreifendes Verständnis legen. Bisher ist allerdings nur wenig bekannt über das Vorwissen der SchülerInnen in Bezug auf Informatiksysteme und über die Wirksamkeit der eingesetzten Unterrichtsmaterialien im Hinblick auf den Erwerb von informatischen Kompetenzen. In der hier beschriebenen Studie mit insgesamt 137 Kindern wurde das Vorwissen von Viertklässlern mit einem Leitfadeninterview ermittelt. Danach wurde eine fünfstündige Unterrichtseinheit zu algorithmischen Grundbausteinen und dem Mikrocontroller Calliope mini durchgeführt. Die Unterrichtseinheit endete mit einer Aufgabenstellung, für die die Kinder das Eingabe-Verarbeitung-Ausgabe-Prinzip (EVA-Prinzip) nachvollziehen und mit dem Mikrocontroller umsetzen mussten. Die Bearbeitung dieser Aufgabenstellung wurde per Screencast festgehalten. Unsere Ergebnisse zeigen, dass ein Großteil der Kinder bestenfalls vage Vorstellungen von Informatiksystemen hat, aber sehr gut in der Lage ist, ein altersangemessenes System wie den Calliope mini zielgerichtet zu programmieren.

Keywords: Grundschule; Informatik; Calliope mini

1 Motivation und Ziele der Studie

Informatiksysteme halten mehr und mehr Einzug in den Lebensalltag aller Menschen, darunter auch SchülerInnen der Primarstufe. Doch im Gegensatz zu Phänomenen der natürlichen Umwelt, über die der Sachkundeunterricht größtenteils aufklärt, sind Systeme der digitalen Welt noch nicht Gegenstand des Unterrichts in der Primarstufe. Im Gegensatz zu anderen Ländern gibt es in Deutschland zur Zeit weder ein eigenes Unterrichtsfach noch eine feste Verankerung dieser Themen in den Lehrplänen der bestehenden Grundschulfächer (vgl. [ST11]). Das Calliope-Mini-Projekt des Ministeriums für Bildung und Kultur (MBK) Saarland ist ein erster Schritt, um dies zu ändern: Seit Anfang 2017 werden Lehrerfortbildungen vom Landesinstitut für Pädagogik und Medien im Saarland angeboten, die den Mikrocontroller Calliope mini als didaktisches Informatiksystem zum Einsatz in der Schule vorstellen und Grundkonzepte zur informatischen Bildung in der Primarstufe vermitteln. Das Gerät soll nach Vorstellung des MBK ab Klassenstufe 3 saarlandweit im

¹ Universität des Saarlandes, Campus, DE-66123 Saarbrücken, [kevin.baum|kerstin.reese|pascal.schmidt|verena.wolf]@uni-saarland.de, [s8nakirs|s8luwach]@stud.uni-saarland.de

Regelunterricht eingesetzt werden [LP17].

Ziel der hier beschriebenen Studie ist es, die Vorkenntnisse von GrundschülerInnen zu beleuchten und Calliope mini auf seine Tauglichkeit bei der Einführung von Algorithmen und der Funktionsweise (reaktiver) Informatiksysteme im Regelunterricht zu untersuchen. Das sind wichtige Kompetenzen, die auch von der GI für den Primarbereich gefordert werden (vgl. [Be18]). Die Forschungsfragen, die hierzu geklärt werden sollen, lassen sich auf zwei wesentliche Hauptfragen A und B einschränken.

- A Gelingt es Kindern der 4. Klasse nach einer dreistündigen Einführung in Algorithmen und die Programmierung des Calliope mini, eigenständig Programme zur Lösung vorgegebener Eingabe-Ausgabe-Zusammenhänge zu entwerfen?

Daraus ergeben sich folgende Teilfragen:

- A1 Nutzen die Kinder zielgerichtet algorithmische Grundbausteine (Eingaben, Ausgaben, Sequenzen) sowie Sensoren und Aktoren?
- A2 Lassen sich typische Missverständnisse oder Verständnisschwierigkeiten identifizieren?

Fragestellung B bezieht sich primär auf das Vorwissen der Lernenden:

- B Erkennen Kinder ein ihnen unbekanntes Informatiksystem als solches (und nicht nur als Spielzeug oder allgemeines technisches Gerät) und wie beschreiben sie ein solches System?

Hier werden drei Teilfragen unterschieden:

- B1 Welche Bestandteile und Funktionsweisen eines beobachteten Informatiksystems nennen die Kinder?
- B2 Beschreiben die Kinder einen zutreffenden kausalen Zusammenhang zwischen Eingabe und Ausgabe?
- B3 Falls die Kinder einen solchen Zusammenhang hergestellt haben: Welche Ursache schreiben die Kinder dem genannten Zusammenhang zu?

In der Studie wurde zunächst das Vorwissen von Viertklässlern ermittelt. Danach wurde eine fünfständige Unterrichtseinheit zu algorithmischen Grundbausteinen und dem Mikrocontroller Calliope mini durchgeführt. Die Unterrichtseinheit endete mit einer Aufgabenstellung, für die die Kinder das Eingabe-Verarbeitung-Ausgabe-Prinzip (EVA-Prinzip) nachvollziehen und mit dem Mikrocontroller umsetzen mussten.

Unsere Ergebnisse zeigen, dass ein Großteil der Kinder bestenfalls vage Vorstellungen von Informatiksystemen hat, aber sehr gut in der Lage ist, ein altersangemessenes System wie den Calliope mini zielgerichtet zu programmieren. Er ermöglicht den Kindern einen ersten Zugang zum EVA-Prinzip und wird erfolgreich von ihnen zur Problemlösung eingesetzt.

2 Organisation der Studie

Die Studie wurde von September bis Dezember 2018 in acht Klassen an fünf saarländischen Grundschulen mit insgesamt 137 Kindern durchgeführt (Tab. 1). Die nachfolgend beschrie-

Nr.	1	2	3	4	5	6	7	8	Σ
Schulcode	1	1	1	2	3	4	4	5	
Anzahl SchülerInnen	20	17	21	18	17	15	18	11	137

Tab. 1: Anzahl der an der Studie teilnehmenden Schulen, Klassen und SchülerInnen

bene Unterrichtseinheit hat einen Umfang von drei Doppelstunden. Unterrichtet wurde im Klassenraum mit einem Team von zwei Personen, wobei die Lehrperson der Klasse zusätzlich anwesend war. Das erforderliche technische Equipment – Calliope mini, Beamer, Laptops – wurde vom Projektteam vorbereitet und mitgebracht. Auf den Laptops wurde mithilfe eines lokalen Web-Servers eine Offline-Version des Microsoft MakeCode-Editors [Mi] betrieben, um von einem Internetzugang unabhängig zu sein. Der Calliope mini war als Download-Ordner des Web-Browsers konfiguriert, so dass beim Klick auf den Knopf „Herunterladen“ sofort der angeschlossene Calliope mini bespielt wurde.

Um die Vorstellungen der Kinder von Informatiksystemen zu untersuchen, wurden Cubelets benutzt [Mo]. Die über Magnete zusammensteckbaren Cubeletmodule hatten sich in Vortests bewährt, weil sie nicht direkt als „Computer“ erkennbar sind. Cubeletmodule sind Blöcke mit unterschiedlichen Funktionen, wie z.B. Motoren oder Sensoren. Damit stellen sie ein abstraktes, programmierbares Informatiksystem dar, das den Kindern recht wahrscheinlich nicht bereits bekannt war.

3 Verlauf der Unterrichtseinheit

Erste Doppelstunde: Interviews, Wortfeldarbeit und AIGo. Zur Exploration des Vorwissens wurden alle SchülerInnen zunächst einzeln mithilfe eines standardisierten Leitfadens interviewt: Nach einer kurzen Demonstration der Funktionsweise eines Cubelet-Aufbaus mit Motor- und Sensorblock – das Erkennen eines Gegenstandes im Nahbereichs des Sensors setzt den Motor für wenige Sekunden in Bewegung – wurden die SchülerInnen aufgefordert, das Gesehene zu beschreiben bzw. zu erklären.

Die im Verlauf der Interviews gesammelten Begriffe wurden auf Kärtchen notiert und im Plenum nacheinander an die Tafel geheftet, ggf. geklärt und anhand verschiedener Kriterien (z.B. Verhalten vs. Bestandteile des Roboters) sortiert. Neben einer gemeinsamen (sprachlichen) Ausgangsbasis wurde dadurch eine erste Sensibilisierung für die im Hinblick auf das EVA-Prinzip notwendige Unterscheidung zwischen Ein- und Ausgaben geschaffen. Die Einführung algorithmischer Grundkonzepte erfolgte zunächst unplugged, um das gleichzeitige Auftreten konzeptioneller und technischer Herausforderungen zu verhindern und um eine Ablenkung durch rein technische Probleme zu verhindern. Dafür wurde das Algorithmenspiel Algo! verwendet, welches lediglich LEGO-Zubehör und Befehlskarten mit Piktogrammen nutzt. Die einfachste Aufgabenvariante besteht darin, eine Spielfigur entlang eines Weges zu steuern, indem der passende Algorithmus mithilfe von elementaren Anweisungen eines begrenzten Befehlssatzes beschrieben wird. Nach einer kurzen Kennenlernphase bearbeiteten die SchülerInnen in Zweiergruppen mit jeweils eigenem

Spielset unterschiedlich komplexe Szenarien (z.B. auch die Umkehraufgabe: Wegebau zu vorgegebenem Algorithmus).

Zweite Doppelstunde: Calliope ohne und mit Strom. In der ersten Hälfte der Doppelstunde lernten die Kinder die Bestandteile der (zunächst noch stromlosen) Minis kennen und ordneten diese den Kategorien „Eingabe“, „Ausgabe“ und „Sonstiges“ zu. Anschließend erkundeten sie die Funktionsweise der Geräte: Hierzu wurden vier unterschiedlich programmierte Sorten von Minis (Kombinationen aus Eingabe (Knopf A gedrückt, geschüttelt) und Ausgabe (Smiley-Folge, Ampel-Farben)) an die SchülerInnen verteilt und jeweils ein prototypischer Vertreter in eine Raumecke platziert. Die SchülerInnen mussten sich nun so im Klassenraum positionieren, dass in den Ecken nur gleich funktionierende Minis (d.h. gleiche Kombination von Ein- und Ausgabe) zusammenkamen. In der anschließenden Besprechung wurde herausgearbeitet, dass sich das unterschiedliche Verhalten der Minis durch unterschiedliche, von Menschen gemachte Programmierung erklärt – damit war eine Motivation zum Kennenlernen der MakeCode-Entwicklungsumgebung gegeben.

Die Präsentation der Entwicklungsumgebung und die Programmierung eines Mini wurden zunächst am Beamer erläutert und anschließend von den SchülerInnen in Zweiergruppen am eigenen Rechner nachvollzogen. Nach erfolgreicher Reproduktion eines zweiten Beispiels durften die Kinder die vorgegebenen Programme verändern und weitere Befehle aus den Bereichen „Grundlagen“, „Eingabe“ und „Musik“ ausprobieren.

Dritte Doppelstunde: Abschlussaufgaben und Projekte. In der dritten Doppelstunde bearbeiteten die Kinder in Zweiergruppen vier Programmieraufgaben mit ansteigendem Schwierigkeitsgrad. Alle Programme sind mit sequenziellen Befehlen lösbar, wie auf der ersten Stufe des Stufenmodells von [HWW17]. Die SchülerInnen wurden nicht darüber informiert, dass es sich um „Testaufgaben“ handelt, um eine möglichst natürliche und ungezwungene Atmosphäre zu schaffen. Im Einzelnen sollten folgende Aufgaben bearbeitet werden:

Aufgabe 1: Reproduktion einer bereits bekannten Programmierung („Auf Knopfdruck A: Smileys anzeigen“).

Aufgabe 2: Transfer durch leichte Variation („Auf Knopfdruck B: Text „CODE“ anzeigen“).

Aufgabe 3: Vorlage eines Calliope mini, dessen Verhalten die Kinder zunächst erkunden und dann in Form eines Programms rekonstruieren sollten („Auf Knopfdruck A: längere Sequenz von unterschiedlichen Ausgaben“).

Aufgabe 4: Aufgabenstellung wie Aufgabe 3, bis auf komplexeres Verhalten des Calliope mini (drei unterschiedliche Inputs, die jeweils unterschiedliche Outputfolgen auslösen).

Die Bearbeitung der Abschlussaufgaben beanspruchte durchschnittlich 50 Minuten. In der restlichen Zeit durften die SchülerInnen selbstständig eigene Projekte realisieren und diese am Ende der Doppelstunde vor der gesamten Klasse präsentieren. Die Auswertung des freien Programmierens ist nicht Teil dieser Veröffentlichung.

ID	Kategoriebezeichnung	Definition	Ankerbeispiel
OK 1	Verhalten		
UK 1.1	kann Verhalten reproduzieren	SchülerIn wird selber aktiv, handelt und reproduziert das gezeigte Verhalten	
UK 1.2	kann Verhalten beschreiben	SchülerIn beschreibt das Beobachtete	„Wenn man die Hand dahin hält, fährt er nach links“, Interview-Nr. 59
OK 2	Begriffe		
UK 2.1	benutzt mystische Begriffe	Aussagen, in denen Begriffe auftreten, die vermenschlichen oder eine höhere Gewalt hinzuziehen	„Es hat Angst, dass es kaputt geht und weiß, dass es weglaufen soll“, Interview-Nr. 61
UK 2.2	benutzt nicht-mystische, aber nicht-technische Begriffe	Aussagen, in denen Begriffe aus der Biologie oder Analogien mit dem menschlichen Körperbau, z.B. Augen, beinhaltet sind	„Die Augen reagieren auf die Hand“, Interview-Nr. 64
UK 2.3	benutzt technische Begriffe	Aussagen, in denen Begriffe, die für ein technisches System angemessen sind, z.B. Räder, Sensoren, beinhaltet sind	„Da ist ein Sensor drin, der Bewegung wahrnimmt“, Interview-Nr. 82
OK 3	Weiterführend/Spezieller		
UK 3.1	gibt Antwort auf „Warum“-Frage	Antwort auf die „Warum“-Frage, die für den Schüler/die Schülerin sinnvoll ist	„Da ist was drin, das dafür sorgt, dass es sich bewegt“, Interview-Nr. 96
UK 3.2	erwähnt, dass ein Mensch das so gebaut/programmiert hat	Aussagen, in denen erwähnt wird, dass ein Mensch das Informatiksystem gebaut/programmiert hat	„Jemand hat sich was ausgedacht und wollte es umsetzen“, Interview-Nr. 93
UK 3.3	verwendet den Begriff „Programmierung“	SchülerIn kennt das Wort und verwendet es im Interview	„Das ist so programmiert“, Interview-Nr. 116
UK 3.4	hat eine Erklärung dafür, was „Programmierung“ ist	Auf Nachfrage: Erklärung des Begriffs „Programmierung“, die für den Schüler/die Schülerin sinnvoll ist	„Man kann ihm sagen, was es machen soll“, Interview-Nr. 103

Tab. 2: Kategoriensystem zur Einordnung der Aussagen in den Interviews

4 Erhebungsinstrumente und Auswertungsmethoden

Während der Interviews zur Vorwissensabfrage wurden Reaktionen und Antworten der Kinder von einem Protokollanten in Form von schriftlichen Notizen festgehalten.

Bei den Programmieraufgaben wurden die Aktionen am Computer per Screencast erfasst. Die Screencasts wurden mit der Software OBS [Op] angefertigt, die automatisch mit dem Editor startete. Während der Bearbeitung der Programmieraufgaben wurden Rückfragen der Kinder beantwortet, ggf. auch Hilfestellungen und Tipps geben – beides wurde mit Zuordnung zu dem/der betreffenden Schüler(in) und Zeitstempel notiert. Ton- oder Videoaufzeichnungen wurden nicht angefertigt.

Die Interviews wurden in Anlehnung an Mayring (vgl. [Ma02]) ausgewertet, da lediglich Notizen zu den Schülerantworten und -reaktionen vorlagen und keine vollständigen Transkriptionen. Tab. 2 zeigt das Kategoriensystem, welches induktiv entwickelt und in Rückkopplung mit seiner Anwendung mehrfach angepasst wurde.

Da die Notizen kurz und die Definitionen und Ankerbeispiele bereits sehr konkret sind, wurde auf Codierregeln verzichtet. Das Kategoriensystem wurde von mehreren Personen zur unabhängigen Auswertung genutzt. Im nächsten Schritt ließen sich die Unterkategorien „mystische“ (UK 2.1), „nicht-technische“ (UK 2.2), „technische“ (UK 2.3) Begriffe, die Antwort auf die „Warum-Frage“ (UK 3.1) und die Erklärung des Begriffs „Programmierung“ (UK 3.4) in Anlehnung an Mayring reduzieren.

Bei der quantitativen Auswertung der Screencasts ist für jedes Team und jede Aufgabe bestimmt worden, ob die Aufgabe vollständig, bis auf kleine Mängel oder gar nicht gelöst wurde. Ebenso wurde die Bearbeitungsdauer erfasst. Aufgabe 4 wurde aufgrund ihrer Komplexität genauer ausgewertet. Um diese Aufgabe erfolgreich zu lösen, musste sie in Teilaufgaben zerlegt werden. Der Erfolg beim Erkennen und Bearbeiten der Teilaufgaben und das Vorgehen der Schülerinnen und Schüler wurden festgehalten.

5 Ergebnisse der Studie

5.1 Ergebnisse der Interviewauswertung

Es lagen 133 Notizen zu Interviews vor, die ausgewertet werden konnten (Tab. 3). Als mystische Begriffe (UK 2.1) wurde z.B. „Jedi-Fahrzeug“ genannt. Es wurde unterstellt, dass die Cubelets „denken“ oder „sich erschrecken“. Zusammenfassend kann gesagt werden, dass diese Gruppe von Kindern die Cubelets als handelnd beschreibt. Neun Kinder haben im Interview mystische Begriffe verwendet.

Die Reduktion der nicht technischen Begriffe (UK 2.2) ergab, dass diese Gruppe von Kindern das eingesetzte Informatiksystem häufig mit Analogien aus der Biologie beschreibt. Die Cubelets haben „Augen“ oder die „Sonne löst etwas aus“. Meist werden Begriffe für den sichtbaren Teil der Sensoren gesucht: „Löcher“, „das Schwarze“ oder „Punkte“. 41 Kinder haben wenigstens einen nicht-technischen Begriff benutzt.

An technischen Begriffen (UK 2.3) wurden meist „Sensor“ oder Analogien („Scanner“, „Bewegungsmelder“) genannt. Die sichtbaren, mechanischen Bestandteile der Cubelets wurden öfter mit Begriffen wie „Räder“ oder „Rollen“ beschrieben. Innerhalb der Cubelets wurden „Kabel“, „Motoren“, „Elektronik“ und „Batterien“ vermutet. Manche Kinder dieser Gruppe erklärten die berührungsfreie Interaktion mit den Cubelets mit „Magneteten“ oder „Wärme“. Einige sagten, dass „Signale“ oder „Daten“ vom Sensor an den Motor geschickt werden. Insgesamt kann der Gebrauch der technischen Begriffe als überwiegend vage bezeichnet werden. 115 Kinder benutzten technische Begriffe.

Als Antwort auf die „Warum“-Frage (UK 3.1) wurde meist genannt, dass das System „so programmiert“ oder etwas „so eingebaut“ sei. Als Erklärung wurde oft nur gesagt, dass es einen „Sensor“ gibt. Die Antworten blieben kurz und kamen über Schlagworte – „Technik“, „Elektronik“, „Signale“ – kaum hinaus. Oft wurde erst auf Nachfrage eine Antwort gegeben. 66 Kinder gaben eine Antwort auf die „Warum“-Frage.

Wenn die Kinder den Begriff „Programmieren“ benutzt haben, wurde nachgefragt, was das sei (UK 3.4). Die Kinder erläuterten, dass „ein Mensch etwas sagt“ oder „etwas eingestellt wird“. Das macht man „am Computer“ und „überträgt“ es auf das Informatiksystem. 24

Schulcode	1	1	1	2	3	4	4	5	Σ
Anzahl Interviews	17	17	21	18	17	15	17	11	133

Tab. 3: Anzahl der verwertbaren Interviews

	bearbeitet	vollst. gelöst	kl. Fehler	nicht gelöst	min	∅	max
Aufg. 1	63	58	4	1	01:00	03:19	19:31
Aufg. 2	59	57	1	1	00:38	04:48	17:59
Aufg. 3	62	47	10	5	03:23	13:38	37:12
Aufg. 4	54	9	0	45	13:18	23:09	36:29

Tab. 4: Anzahl der bearbeiteten Aufgaben, Erfolg und Bearbeitungszeiten

Kinder hatten eine eher vage Vorstellung davon, was Programmieren ist. Kein Kind schien von eigenen Erfahrungen berichten zu können.

5.2 Ergebnisse aus der Screencast-Auswertung

Nach der Erhebungsphase lagen 63 auswertbare Screencasts mit insgesamt mehr als 50 Stunden Videomaterial vor. Tab. 4 zeigt den nach Aufgaben aufgeschlüsselten Erfolg der Bearbeitung sowie die Bearbeitungsdauer.

Festzustellen ist, dass bei den ersten drei Aufgaben jeweils über 90% der Gruppen eine korrekte Lösung oder eine Lösung mit nur kleinen Fehlern gefunden haben. Zur erfolgreichen Bearbeitung der 3. Aufgabe mussten die Teams ein komplexeres Verhalten des Calliope mini nachbilden. Dazu musste das Verhalten der Vorlage genau beobachtet und Schritt für Schritt programmiert werden. Nahezu 75% der Teams konnten diese Aufgabe vollständig lösen, während nur ca. 10% keine Lösung gelang.

Bei der 4. Aufgabe mussten mehrere Eingabe-Ausgabe-Zusammenhänge erkannt und implementiert werden. Besonders gut gelang es den Kindern, eine rein visuelle Sequenz zu erkennen und zu reproduzieren: Zwei Drittel konnten die korrekte Reihenfolge für das Lauflicht auf dem LED-Feld programmieren.

Zur Bestimmung der Bearbeitungsdauer der einzelnen Aufgaben (vgl. Abb. 1) wurde die Zeit gemessen, die entsprechend der Screencasts an der Lösung der Aufgabe verbracht wurde. Da die Aufgaben ohne Vorplanung lösbar waren und im Unterricht keine Planungszeit ohne Aktivität am Computer beobachtet werden konnte, gehen wir davon aus, dass die ermittelten Bearbeitungszeiten in Tab. 4 ausreichend genau sind. Zu erkennen ist, dass die Bearbeitungsdauer mit ansteigendem Schwierigkeitsgrad zunehmend stärker variiert. Das lässt vermuten, dass die erste Aufgabe als Einstieg den meisten Kindern gerecht wurde.

Die in einzelnen Fällen sehr geringe Bearbeitungszeit bei Aufgabe 2 ergibt sich daraus, dass nur wenige Teams den Baustein „Zeige Zeichenfolge“ benutzt haben. Die meisten Teams haben jeden Buchstaben einzeln auf dem LED-Feld gezeichnet, womit diese Umsetzung länger dauerte.

Gut 87% der Teams waren während (mehr oder minder) der gesamten Bearbeitungszeit mit den Aufgaben beschäftigt. Die Teams, die nicht an der Lösung gearbeitet haben, schauten sich an anderer Stelle des Editors um. Kein Team hat den Editor verlassen, andere Programme aufgerufen oder die Arbeit am Computer komplett eingestellt.

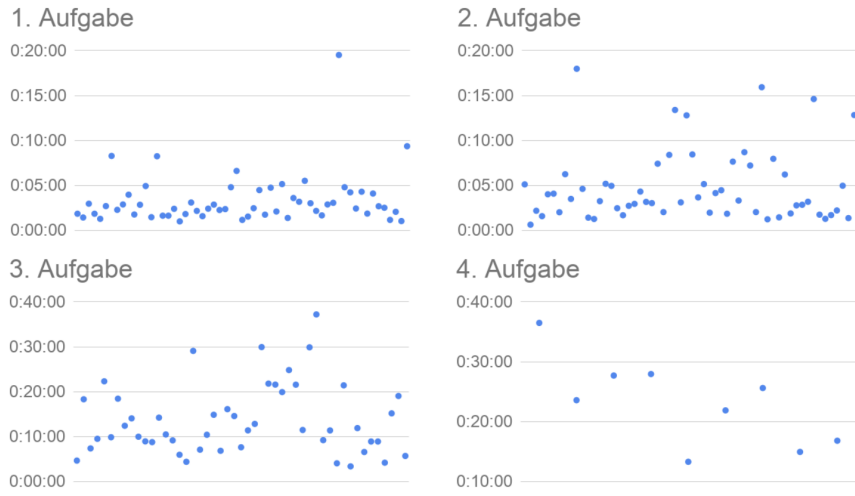


Abb. 1: Dauer der erfolgreichen Bearbeitung für jede Aufgabe

Die 4. Aufgabe bestand aus drei Teilaufgaben, die grundsätzlich in beliebiger Reihenfolge abgearbeitet werden konnten:

Teilaufgabe 1: Dauerhaft blinkendes Muster auf dem LED-Feld

Teilaufgabe 2: Wenn Knopf A gedrückt wird, werden 3 verschiedene Töne gespielt

Teilaufgabe 3: Wenn Knopf B gedrückt wird, werden 3 verschiedene Töne gespielt und der Ampelfarbverlauf (Rot, Gelb, Grün) dargestellt

Fast alle Teams bearbeiteten als erstes Teilaufgabe 1, die auch am häufigsten korrekt gelöst worden ist (gut 60% aller Teams). Bei den Teilaufgaben 2 und 3 sank der Anteil der erfolgreichen Teams auf ca. ein Viertel.

5.3 Beobachtungen während der Arbeit an den Abschlussaufgaben

Gelegentlich fehlten Teile des zu entwickelnden Algorithmus, wie z.B. das Ausschalten des LED-Felds am Ende des ersten Programms. Insbesondere bei Aufgabe 3 fehlte oft der finale Ton, da der Calliope mini nicht sehr laut piept. Es fiel einigen Kindern schwer, selbstständig neue Bausteine, wie z.B. den „dauerhaft“-Baustein für Aufgabe 4 oder den „Wenn Knopf B gedrückt“-Baustein für Aufgabe 2, zu integrieren. Wir vermuten, dass dies an der Kürze der Arbeit mit dem Editor lag und Probleme dieser Art mit zunehmender Übung seltener auftreten dürften.

Nur wenige Kinder verwendeten bei Aufgabe 3 auf Anhieb auch das Ausschalten des LED-Felds, um es zum Blinken zu bringen. Fast alle versuchten das Blinken mit dem „Pausiere“-Baustein umzusetzen. Dies wurde möglicherweise durch die Programmierung der Ampel in der Einführung begünstigt, bei der man diesen Baustein benötigt. Zudem könnte der Bezeichner dazu geführt haben, dass die Kinder in den „Pausiere“-Baustein eine andere Semantik hinein interpretierten ("tue eine Zeit lang nichts"), von der sie auch nur schwer abzubringen waren. Die Verwendung eines leeren „Zeige LEDs“-Bausteins in Verbindung mit dem „Pausiere“-Baustein überzeugte die Kinder deutlich mehr.

6 Fazit und abschließende Worte

Die Ergebnisse der Studie zeigen einerseits die geringen Vorkenntnisse der interviewten SchülerInnen und offenbaren zugleich das Potential kindgerechter reaktiver Informatiksysteme – hier in Form des Calliope mini – für das Erlernen informatischer Grundprinzipien. Ein Großteil der 137 TeilnehmerInnen verschiedener Schulen in unterschiedlichen Einzugsgebieten hatte bislang allenfalls rudimentäres Vorwissen über Informatiksysteme. Eine adäquate Vorstellung der Funktionsweise des EVA-Prinzips konnte nahezu überhaupt nicht festgestellt werden.

Die Ergebnisse der abschließenden Programmierarbeit zeigen, dass die Kinder mit den einfachen EVA-Problemstellungen, wie sie in den Aufgaben 1-3 zu lösen waren, kaum Probleme hatten. Dies zeigt auch, dass sie im Laufe der Unterrichtseinheit die Kompetenz erworben haben, mit elementaren Programmierelementen und dem Editor sinnvoll umzugehen. Die 4. Aufgabe erforderte eine Zerlegung in überschaubare Teilprobleme, also eine durchaus fortgeschrittene Kompetenz des "computational thinking". Insofern ist es bewerkenswert, dass neun Teams (ca. 14%) diese Aufgabe vollständig korrekt lösen konnten und fast alle anderen Teams immerhin korrekte Teilergebnisse bei der Wahl der Ein-Ausgabe-Bausteine und der Definition der Sequenzen angeben konnten.

Insgesamt ziehen wir hieraus das Fazit, dass SchülerInnen der 4. Klasse in der Lage sind, einfache Beispiele für das EVA-Prinzip zu verstehen und in einem konkreten Informatiksystem zu implementieren – wie es auch [Sc01] aus kognitionspsychologischer Sicht bestätigt. Der Calliope mini stellte sich als gutes Hilfsmittel zur Entmystifizierung solcher Systeme heraus und ist den Ergebnissen der Studie nach zu urteilen als didaktisches Informatiksystem zum Einsatz in der Primarstufe geeignet. Zu sehr ähnlichen Schlussfolgerungen kommt auch eine Studie von Murmann et al. (vgl. [Mu18]), in der ebenfalls Unterrichtseinheiten für Calliope mini entworfen und durchgeführt worden sind. Während der Bremer Fokus auf Beobachtungen und Befragungen der Kinder lag, die mit den Calliope minis arbeiteten, beinhaltet unsere Studie auch die Auswertung eines Abschlusstests und fokussiert damit auch die Performanz der SchülerInnen. Eine weitere Studie mit der Programmiersprache Scratch findet sich in [FGH17].

Abschließend möchten wir einige persönliche Eindrücke schildern: Viele Lehrkräfte waren im Vorfeld skeptisch und zeigten gegenüber den Geräten Berührungängste. Manche konnten sich kaum vorstellen, dass ihre SchülerInnen im Umgang mit den Minicomputern

etwas Sinnvolles lernen könnten. Die Unterrichtseinheit hat dieses Bild deutlich verändert, weil sie den Kindern Spaß machte und ihre Ergebnisse die Erwartungen der Lehrkräfte schon nach kurzer Zeit deutlich übertrafen – eine Erkenntnis, die bei möglichst vielen Grundschullehrkräften erreicht und gefestigt werden muss.

Danksagung Wir danken Ludger Humbert (Universität Wuppertal), der uns in der Vorbereitungsphase für die Studie unterstützt hat. Ebenso danken wir den beteiligten LehrerInnen und SchülerInnen für ihre Bereitschaft, an der Studie teilzunehmen.

Literatur

- [Be18] Best, A.; Borowski, C.; Büttner, K.; Freudenberg, R.; Fricke, M.; Haselmeier, K.; Herper, H.; Hinz, V.; Humbert, L.; Müller, D.; Thomas, M.: Kompetenzen für informatische Bildung im Primarbereich. LOG IN 38/189/190, S. 1–36, 2018.
- [FGH17] Funke, A.; Geldreich, K.; Hubwieser, P.: Analysis of scratch projects of an introductory programming course for primary school students. In: IEEE Global Engineering Education Conference (EDUCON). S. 1229–1236, Apr. 2017.
- [HWW17] Hoffmann, S.; Wendlandt, K.; Wendlandt, M.: Algorithmisieren im Grundschulalter. In: Informatische Bildung zum Verstehen und Gestalten der digitalen Welt. Gesellschaft für Informatik, Bonn, S. 73–82, 2017.
- [LP17] LPM: Bildungsminister Commerçon stellt Landeskonzept Medienbildung und Förderprogramm für Medienschulen vor, 2017, URL: https://www.lpm.uni-sb.de/typo3/index.php?id=2092&tx_ttnews%5Btt_news%5D=679&cHash=fc7e178c61bb18f79092d46e441cccf0&L=0, Stand: 18.05.2019.
- [Ma02] Mayring, P.: Einführung in die qualitative Sozialforschung. Beltz, 2002.
- [Mi] Microsoft: MakeCode, URL: makecode.calliope.cc, Stand: 18.05.2019.
- [Mo] Modular Robotics: Cubelets, URL: www.modrobotics.com, Stand: 18.05.2019.
- [Mu18] Murmann, P. D. L.; Schelhowe, P. D. H.; Bockermann, D. I.; Engelbertz, S.; Illginnis, S.; Moebus, A.: Calliope mini - Eine Explorationsstudie im pädagogisch-didaktischen Kontext, Abschlussbericht, 2018, URL: <https://elib.suub.uni-bremen.de/edocs/00106848-1.pdf>, Stand: 18.05.2019.
- [Op] Open Broadcaster Software: OBS Studio, URL: <https://obsproject.com>, Stand: 18.05.2019.
- [Sc01] Schwill, A.: Ab wann kann man mit Kindern Informatik machen? - Eine Studie über informatische Fähigkeiten von Kindern. In: Informatikunterricht und Medienbildung. INFOS '01, GI, S. 13–30, 2001.
- [ST11] Starruß, I.; Timmermann, B.: Informatische Bildung in Deutschland. LOG IN 31/1, S. 49–59, 2011.