Ira Diethelm (Hrsg.): Informatische Bildung zum Verstehen und Gestalten der digitalen Welt,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 1

Willkommen im Programmierzirkus – Ein Programmierkurs für Grundschulen

Katharina Geldreich[1] Alexandra Funke[2] Peter Hubwieser[3]

Abstract: Informatik soll in den kommenden Jahren immer stärker in die frühe Bildung integriert werden und somit den Weg in die Grundschulen sowie die Kindergärten finden. Dafür ist es erforderlich, Methoden und Inhalte zu erforschen, die für Schülerinnen und Schüler der Primarstufe geeignet sind. Auf dieser Basis haben wir einen dreitägigen Kurs, den Programmierzirkus, für Dritt- und Viertklässler entwickelt, der den Kindern einen Einblick in die Programmierung geben soll. Die Kinder sind zwischen acht und zehn Jahren alt und befinden sich kurz vor dem Übergang an die weiterführende Schule. Im Jahr 2016 nahmen 58 Schülerinnen und Schüler erfolgreich an dem Kurs teil. Weitere Durchführungen sind im Juni und Juli 2017 mit 63 Dritt- und Viertklässlern geplant. Dieser Beitrag beschreibt den Aufbau der einzelnen Kurstage und die verwendeten Lehr-Lern-Methoden. Weiterhin werden die Aufzeichnungen und Auswertungen des Programmierzirkus beschrieben, bei dem eine Vielzahl an Methoden zum Einsatz kam.

Keywords: Programmieren; informatische Bildung; Grundschule; CS unplugged; Scratch

1 Einleitung

Das Fach Informatik wird an Schulen und Universitäten mit verschiedenen Herausforderungen konfrontiert. Beispielsweise haben Schülerinnen und Schüler sowie Studentinnen und Studenten unterschiedliche Fehlvorstellungen und Vorurteile bezüglich des Faches [FBH16], die sich zum Großteil bereits in der Kindheit entwickeln und manifestieren [Pr14]. Um zu verhindern, dass die Lernenden solche meist negativen Einstellungen entwickeln, ist ein aktueller Forschungsansatz, informatische Konzepte wie Programmierung bereits in die Grundschule zu integrieren. Dies ermöglicht den Kindern, eigene Erfahrungen mit Informatik und neuen digitalen Medien zu sammeln. Sie lernen, den Computer nicht nur als Benutzer einzusetzen, sondern mit dessen Hilfe kreativ zu werden [AGE14]. Dieser Rollenwechsel in Kombination mit interessanten Erfahrungen könnte die Selbstwirksamkeit der Schülerinnen und Schüler im Allgemeinen und speziell bezüglich der Informatik und Programmierung erhöhen [To15]. Zeitgleich dazu steigt die Diskussion über die Notwendigkeit der Informatik in der frühen Bildung stetig. Während mehrere Länder bereits das Fach

[1] Technische Universität München, School of Education, Professur für Didaktik der Informatik, Arcisstr. 21, 80333 München, katharina.geldreich@tum.de
[2] Technische Universität München, School of Education, Professur für Didaktik der Informatik, Arcisstr. 21, 80333 München, alexandra.funke@tum.de
[3] Technische Universität München, School of Education, Professur für Didaktik der Informatik, Arcisstr. 21, 80333 München, peter.hubwieser@tum.de

Informatik in die Grundschulcurricula integriert haben (z. B. Großbritannien [Br14] und Australien [FVF14]), hat Deutschland noch keine verbindlichen Richtlinien für den Umgang mit diesen Themen entwickelt. Um herauszufinden, welche Lehrmethoden und Inhalte für Kinder im Grundschulalter geeignet wären, haben wir einen einführenden Programmierkurs für Schülerinnen und Schüler der Grundschule entworfen.

2 Kurskonzept

Der Kurs wurde zunächst für die vierte Klasse konzipiert und hat das Ziel, den Schülerinnen und Schülern innerhalb von drei Tagen einen Einblick in die Programmierung zu geben. Dabei sollen sie ganz grundsätzlich verstehen, wie ein Computerprogramm funktioniert, und darüber hinaus befähigt werden, sich in der Programmierumgebung Scratch zurechtzufinden und dort eigene multimediale Projekte zu programmieren.

Das Thema „Zirkus“ zieht sich aus mehreren Gründen als durchgängiges Motiv durch den Kursablauf, die Aufgaben und die Materialien. Ein wichtiger Aspekt bei der Wahl dieses Themas war die Motivation der Schülerinnen und Schüler. Da die meisten Kinder persönliche Erfahrungen mit einem Zirkusbesuch verbinden können, ist die Themenwahl ein erster Schritt, um die Kursinhalte lebendig und interessant zu vermitteln und dabei an die Lebenswirklichkeit der Schülerinnen und Schüler anzuknüpfen. Des Weiteren hofften wir, Mädchen und Jungen mit diesem Thema gleichermaßen anzusprechen, da weder Mädchen noch Jungen vorrangig damit assoziiert werden. Zusätzlich bietet ein Zirkus vielfältige Aufgabenanlässe, in denen die Handlungen von Tieren und Personen simuliert werden können.

An jedem der drei Kurstage haben wir vier Stunden mit der Klasse verbracht, die nach und nach ein detaillierteres Bild vom Programmieren erhalten sollte. Am Ende des Kurses sollten alle Schülerinnen und Schüler dazu fähig sein, mit der Programmierumgebung Scratch zu arbeiten und algorithmische Kontrollstrukturen anzuwenden und zu kombinieren. Da man in der Grundschule prinzipiell mit einer hohen Variabilität von Schülerleistungen, Lernvoraussetzungen und Lerntypen konfrontiert ist, haben wir uns für eine Kombination verschiedener Lehr- und Arbeitsmethoden entschieden. Einen schematischen Überblick über den Ablauf, die Inhalte und die Ziele des Kurses zeigt Abb. 1.

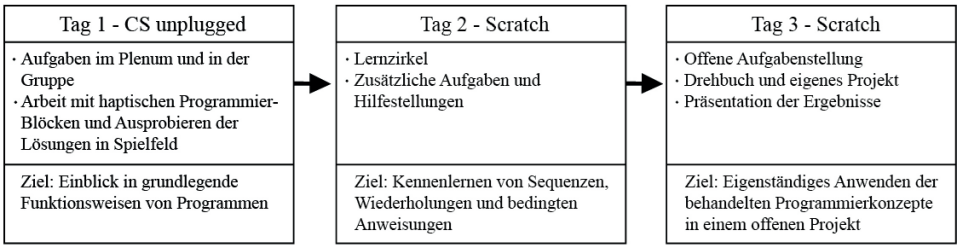

Abb. 1: Ablauf, Inhalte und Ziele des Kurses

Stets im Vordergrund stand durchgängig das Prinzip des aktiven Lernens [PI69] [Vo97]. Die Schülerinnen und Schüler sollten die Möglichkeit haben, sich eigentätig und aktiv mit den Lerngegenständen auseinanderzusetzen. Des Weiteren sollte die Möglichkeit gegeben sein, den informatischen Konzepten auf unterschiedliche Art und Weise zu begegnen [Br66].

2.1 Aufbau des Kurses

Tag 1. Da die meisten Schülerinnen und Schüler keinerlei Vorwissen im Bereich Programmierung oder allgemein in Informatik mitbrachten, war das Ziel des ersten Tages, ihnen eine Grundidee über die Funktionsweise eines Computerprogramms zu geben. Sie sollten begreifen, dass Programme eine bestimmte Aufgabe bewältigen, indem sie präzisen und klaren Anweisungen folgen. Um die Kinder nicht zu überfordern, wurden diese grundlegenden algorithmischen Konzepte „unplugged“ [CS17], das heißt ohne die Verwendung von Computern, eingeführt. Um zu lernen, komplexe Aufgaben in kleinere Teile zu zerlegen, bearbeiteten die Kinder eine Vielzahl von Übungen, in denen Vorgänge in eindeutige Anweisungen umgewandelt werden mussten. Verschiedene einfache Aufgaben wurden zunächst gemeinsam im Plenum gelöst, im Anschluss daran wurden verhältnismäßig komplexere Aufgaben in Kleingruppen bearbeitet. Dabei „programmierten“ die Schülerinnen und Schüler zuerst die Lehrkraft und im Anschluss daran sich gegenseitig. Um das Zirkusthema aufzugreifen, mussten in den Aufgaben beispielsweise vermisste Gegenstände gefunden oder entlaufene Tiere eingefangen werden. Sobald eine Aufgabe gelöst wurde, konnte man die Lösung in einem im Zimmer aufgebauten Spielfeld überprüfen (Abb. 2, rechts). Für die Bearbeitung der Aufgaben fertigten wir haptische Programmierblöcke an, die optisch den Programmierbefehlen in Scratch entsprechen (Abb. 2, links).

Abb. 2: Ein Schüler arbeitet mit den haptischen Scratchblöcken (links); die fertigen Programme werden im Parcours getestet (rechts)

Da die ausgedruckten und laminierten Blöcke mit Magneten und Klettverschlüssen versehen sind, kann mit ihnen sowohl an der Tafel als auch auf großen Filzbahnen gearbeitet werden. Zum einen konnten die Kinder auf diesem Wege ihre Programme ohne Probleme verändern

und im Raum transportieren, zum anderen erhofften wir uns, den Übergang zur Weiterarbeit in Scratch zu erleichtern.

Tag 2. Am zweiten Tag sollten die Schülerinnen und Schüler lernen, einfache Programme am Computer zu erstellen. Da wir eine kinderfreundliche Programmierumgebung verwenden wollten und um mögliche Frustrationen durch syntaktische Fehler zu vermeiden, entschieden wir uns für die visuelle Programmiersprache Scratch [Ma10]. Wir entwickelten einen Lernzirkel mit zunehmend schwierigen Stationen, in dem die Grundfunktionen von Scratch nacheinander behandelt wurden und die jedes Kind in seinem eigenen Tempo bearbeiten konnte. Ausgehend von Fragen der Bedienung führten die erstellten Zirkelkarten (Abb. 3) über einfachen Sequenzen hin zu Kontrollstrukturen wie Wiederholungen und bedingte Anweisungen. An jeder Station wurde die entsprechende Funktion zunächst schrittweise erklärt und die Schülerinnen und Schüler wiederholten jeden Schritt an ihrem eigenen Rechner. Im Anschluss daran bearbeiteten sie eine dazu passende Aufgabe, bei der die eingeführte Funktion in einem anderen Kontext oder leicht abgewandelt verwendet werden musste. Um die heterogene Gruppe gleichermaßen zu unterstützen, standen zusätzliche Aufgaben sowie hilfreiche Tipps für die komplexeren Aufgabenstellungen bereit.

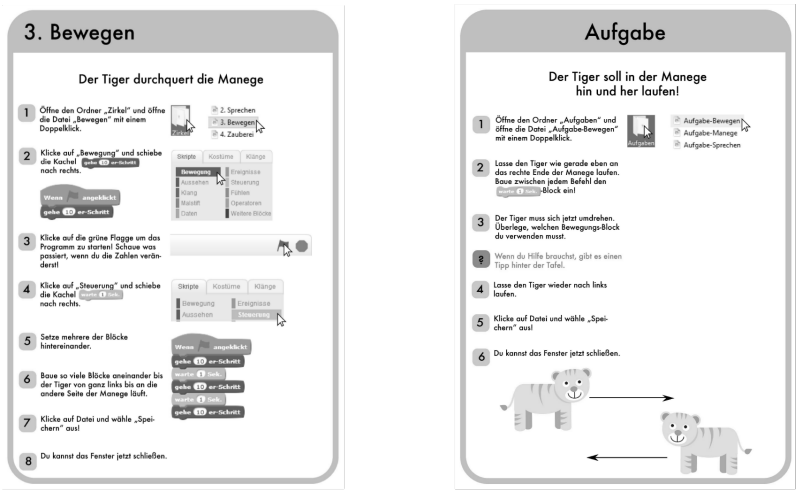

Abb. 3: Vorderseite (links) und Rückseite (rechts) einer Karte des Scratchzirkels

Tag 3. Ziel des dritten Tag war es, herauszufinden, was die Schülerinnen und Schüler gelernt hatten und ob sie das Gelernte auch in einer offeneren Aufgabe anwenden können. Darüber hinaus wollten wir den Kindern eine Möglichkeit bieten, kreativ und selbstbestimmt zu arbeiten. Jedes Kind sollte seine eigene Zirkusgeschichte in einem kurzen Drehbuch (Abb. 4, links) beschreiben und daraufhin in Scratch umsetzen. Da wir trotz der offenen Aufgabenstellung eine Vergleichbarkeit der Schülerprojekte gewährleisten wollten, legten wir einige Anforderungen fest, die erfüllt werden mussten. Die Programme sollten a) mehr als eine Figur beinhalten, b) die Figuren bei der Ausführung bewegen, c) mindestens eine

Wiederholung ausführen und d) mindestens eine bedingte Anweisung enthalten. Nach dem Erfüllen dieser Anforderungen konnten die Schülerinnen und Schüler ihre Arbeit ohne weitere Richtlinien fortsetzen. Ihnen stand es frei, mit Scratch zu experimentieren und ihre eigenen Zirkusgeschichten zu erfinden sowie diese im Anschluss umzusetzen (Abb. 4, rechts). Am Ende des Tages präsentierte jedes Kind seine Zirkusgeschichte der Klasse und hatte Gelegenheit, das Vorgehen und mögliche Schwierigkeiten zu kommentieren.

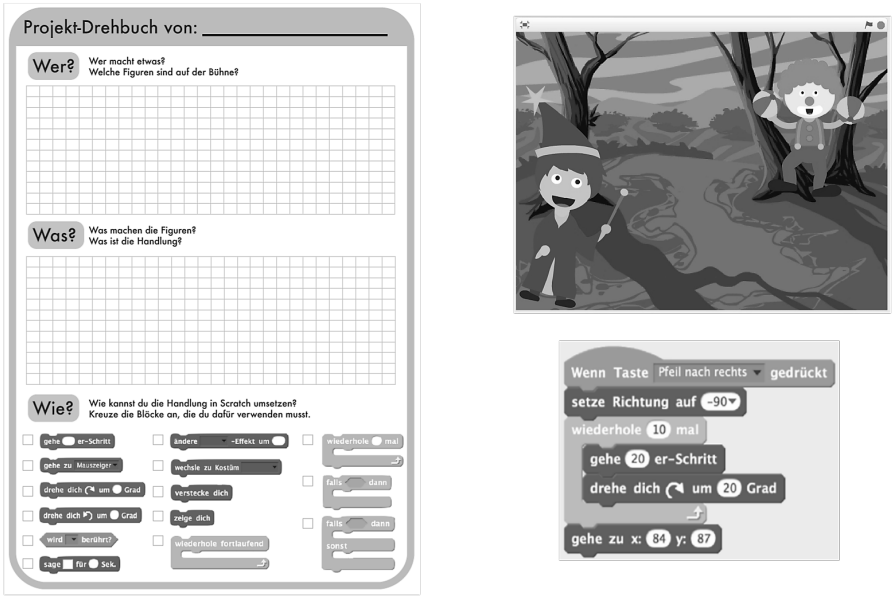

Abb. 4: Drehbuch für die Projekte (links) und das Projekt eines Kindes (rechts)

2.2 Aufzeichnung des Kurses

Um das Kurskonzept und die Effektivität des Kurses zu testen, wurden alle Kursdurchläufe mit einem Mixed-Methods Ansatz aufgezeichnet. Dafür kamen folgende Methoden kombiniert zum Einsatz:

Videografie. Um die Interaktionen der Kinder untereinander und mit der Lehrkraft sowie das Verhalten der Kinder allgemein zu untersuchen, wurden die kompletten Kurse mit vier Kameras aufgezeichnet. Für die entsprechende Tonaufzeichnung kamen ein Ansteckmikrophon für jede Lehrkraft und Richtmikrophone zum Einsatz.

Gruppeninterviews und Fragebögen. Wir nutzten eine Vielzahl an Interview- und Reflektionsmethoden, um einen Einblick in das Vorwissen der Schüler zu bekommen, darüber, welche Vorstellungen sie vom Programmieren haben und wie sie sich vor bzw. nach den Kurstagen fühlten.

Bildschirm- und Audioaufzeichnung. An den Kurstagen zwei und drei wurden die Bildschirme der Kinder mitgeschnitten. Die Sprache wurde über die internen Mikrophone der Laptops aufgenommen. Mit dieser Methode kann ein Einblick in die Arbeitsweise der Kinder erhalten werden.

Scratch-Ergebnisse. Alle Programmierergebnisse der Kinder vom zweiten und dritten Tag wurden gespeichert, um sie im Nachhinein genauer analysieren zu können.

3 Beobachtungen und Auswertungen

Basierend auf der Vielzahl an Daten, die bei der Kursdurchführung erhoben wurden, führten wir verschiedene Auswertungen durch, um einen tieferen Einblick in die Arbeit der Kinder zu erlangen.

Mithilfe eines weiterentwickelten Code-Systems wurden alle 127 Scratch-Ergebnisse der Kinder des dritten Kurstages analysiert [FGH17]. Dabei fanden wir heraus, dass die Projekte vorrangig in drei Projekttypen unterteilt werden können: Spiel, Geschichte und Animation. Diese unterscheiden sich nicht nur in Art und Anzahl der verwendeten Blöcke, sondern auch in der Verwendung der Figuren im Programm. Annähernd alle Kinder erfüllten in ihren Projekten die von uns vorgegebenen Pflichtelemente (siehe 2.1 Tag 3).

Mit Hilfe von Reflektionsbögen, die am Ende jedes Kurstages von den Kindern ausgefüllt wurden, erhielten wir Einblicke, wie sie den Kurs empfanden. Die Bögen dienten zudem dem Erfassen des Vorwissens sowie den aufgetretenen Schwierigkeiten. Alles in allem hatte der Großteil der Kinder an allen Tagen viel Freude – vor allem an den Tagen, an denen mit den Computern gearbeitet wurde. Bei den CS unplugged Aufgaben empfanden die Kinder es als schwierig, den Kursleiter zu "programmieren". Doch nachdem diese erste Hürde gemeistert wurde, fiel den Kindern das Programmieren immer leichter. Dies lässt sich auch gut anhand der Videoaufzeichnungen beobachten. Am meisten Spaß hatten die Kinder am ersten Tag beim Ausprobieren der Lösungen im Spielfeld. Die Antworten des zweiten Tages waren deswegen für uns interessant, weil diese aufzeigen, welche Zirkelaufgaben besonders herausfordernd für die Kinder waren. Anhand dessen kann der Kurs entsprechend verändert werden. Alle Kinder würden gerne weiterhin in der Schule mit Scratch programmieren und etwa 90% wollen dies auch zuhause tun.

Die Bildschirm- und Tonaufzeichnungen der Laptops ermöglichen die Untersuchung des Vorgehens beim Programmieren. In Kombination mit den Videoaufzeichnungen lassen sich Stellen identifizieren, an denen die Kinder auf Probleme gestoßen sind. Diese sind bei den Kindern sehr individuell. Im nächsten Schritt folgt die Untersuchung, ob man verschiedene Programmiertypen unterscheiden kann.

4 Ausblick und zukünftige Arbeiten

Anhand der Beobachtungen und Auswertungen wurden die Materialien, das Kurskonzept und die Aufzeichnungsmethodik evaluiert und überarbeitet. Am Ende des Schuljahres 2016/17 werden mit dem überarbeiteten Kurskonzept vier weitere Durchläufe mit dritten und vierten Klassen durchgeführt. Anschließend sollen die erhaltenen Daten, wie am Beispiel der Daten aus 2016, ausgewertet und mit denen des Vorjahres verglichen werden.

Parallel arbeiten wir an einem Fortbildungskonzept, das Grundschullehrerinnen und -lehrern ohne einschlägiges Vorwissen ermöglichen soll, den Kurs an ihrer Schule zu halten. Fernziel ist hierbei, allen Viertklässlern des Landkreises Berchtesgadener Land die Teilnahme an einem Programmierkurs und somit das Erlernen grundlegender Programmierprinzipien zu ermöglichen.

5 Inhalte des Workshops

Nach einer kurzen thematischen Einführung in das Themengebiet „Programmieren in der Grundschule“ können sämtliche Kursmaterialien selbstständig ausprobiert werden. Darüber hinaus werden verschiedene Schülerergebnisse ausgestellt, anhand derer die Auswertung des Kurses aufgezeigt wird.

Literaturverzeichnis

[AGE14] Armoni, Michal; Gal-Ezer, Judith: Early computing education. ACM Inroads, 5(4):54–59, 2014.

[Br66] Bruner, Jerome: Toward a theory of instruction. Harvard University Press, Cambridge, 1966.

[Br14] Brown, Neil C. C.; Sentance, Sue; Crick, Tom; Humphreys, Simon: Restart: the resurgence of computer science in UK schools. ACM Transactions on Computing Education, 14(2):1–22, 2014.

[CS17] CSUnplugged.org: , Computer Science Unplugged. `csunplugged.org`, March 2017.

[FBH16] Funke, Alexandra; Berges, Marc; Hubwieser, Peter: Different Perceptions of Computer Science. In: 2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE). IEEE, S. 14–18, 2016.

[FGH17] Funke, Alexandra; Geldreich, Katharina; Hubwieser, Peter: Analysis of Scratch Projects of an Introductory Programming Course for Primary School Students. In: Proceedings of the 2017 IEEE Global Engineering Education Conference (EDUCON). IEEE, S. 1233–1240, 2017.

[FVF14] Falkner, Katrina; Vivian, Rebecca; Falkner, Nickolas: The Australian Digital Technologies Curriculum: Challenge and Opportunity. In: Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148. ACE '14, Australian Computer Society, Inc, Darlinghurst, Australia, Australia, S. 3–12, 2014.

[Ma10] Maloney, John; Resnick, Mitchel; Rusk, Natalie; Silverman, Brian; Eastmond, Evelyn: The Scratch Programming Language and Environment. ACM Transactions on Computing Education, 10(4):16:1–16:15, 2010.

[PI69] Piaget, Jean; Inhelder, Bärbel: The psychology of the child. Basic Books, 1969.

[Pr14] Prottsman, Kiki: Computer science for the elementary classroom. ACM Inroads, 5(4):60–63, 2014.

[To15] Topi, Heikki: Gender imbalance in computing. ACM Inroads, 6(4):22–23, 2015.

[Vo97] Vollmers, Burkhard: Learning by doing - Piagets konstruktivistische Lerntheorie und ihre Konsequenzen für die pädagogische Praxis. International Review of Education, 43:73–85, 1997.