

Ulrike DREHER, Landau & Stephanie SCHULER, Landau

## **Computational Thinking in der Grundschule – Erprobung einer Lernumgebung mit dem digitalen Werkzeug Blue-Bot**

Spätestens mit dem Strategiepapier „Bildung in der digitalen Welt“ (KMK, 2017) ist der Bedarf an Lernumgebungen, die den Umgang mit digitalen Werkzeugen fördern und damit digitale Bildung ermöglichen, hinreichend begründet. Die ICILS-Studien 2018 zeigen zudem auf, dass das Computational Thinking (CT) bei deutschen Schüler\*innen im Sekundarbereich durchaus ausbaufähig ist (Eickelmann et al., 2019). Studien mit jüngeren Kindern machen außerdem deutlich, dass das CT bereits in der Grundschule gefördert werden kann (z. B. Caballero-Gonzalez et al., 2019). Die Förderung des CT kann dabei nicht nur im Sachunterricht, sondern auch im Mathematikunterricht erfolgen, da viele Aufgaben auch mathematische Kompetenzen wie das räumliche Vorstellungsvermögen oder die geometrische Begriffsbildung fokussieren können. Im Rahmen des Design-Based-Research-Ansatzes (z.B. Prediger et al., 2012) wurde im Rahmen der vorgestellten Studie eine Lernumgebung für Grundschul Kinder ab Klasse 3 mit dem digitalen Werkzeug Blue-Bot entwickelt und im Hinblick auf die Förderung von Teilkomponenten des CT evaluiert.

### **Theoretischer Hintergrund**

CT wird in der Literatur nicht eindeutig definiert. Viele Definitionen lehnen sich aber an Wing (2006) an, die CT als grundlegende Kompetenz, nicht nur beim Programmieren, sondern auch beim Lösen alltäglicher Probleme, ansieht:

„Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child’s analytical ability.“  
(Wing, 2006, S. 33)

Die vorliegende Studie orientiert sich inhaltlich an den fünf Teilkomponenten nach Angeli et al. (2016, S. 50), wobei für die Arbeit mit dem einfachen digitalen Werkzeug Blue-Bot nur Teilkomponenten 2 bis 4 relevant sind:

- **Abstrahieren:** Irrelevante Eigenschaften müssen ausgeschlossen und relevante beibehalten werden.
- **Umgang mit Algorithmen/algorithmisches Denken:** Es muss eine Schritt-für-Schritt-Anleitung für das zu lösende Problem gefunden werden. Dies umfasst das Sequenzieren (2.1), also das Finden der Teilschritte und deren korrekte Abfolge, und den Kontrollfluss (2.2), also das parallele Überprüfen der Abfolge der Teilschritte.
- **Debugging:** Fehler in einem Algorithmus finden und diese beheben.

- Generalisieren: Verallgemeinern von Lösungen, sodass sie auch auf andere Probleme übertragen werden können. Hierbei spielt das Erkennen von Mustern eine Rolle.
- Dekomposition: Zerlegen komplexer Probleme in Teilprobleme, um sie leichter lösen zu können.

## **Forschungsstand**

Die Studienlage ist in Bezug auf die Förderung des CT in der Grundschule überschaubar. In einer Trainingsstudie konnte beispielsweise gezeigt werden, dass einfache Bodenroboter wie der Bee-Bot oder Blue-Bot zur Förderung des CT eingesetzt werden können (Caballero-Gonzalez et al., 2019). In der Studie sollten Erstklässler\*innen den Bee-Bot zu einem bestimmten Zielpunkt auf einem Gitterraster steuern oder Fehler in vorgegebenen Befehlsfolgen finden. In einem Prä-Post-Test-Design konnte gezeigt werden, dass Lernzuwächse bei den Komponenten Sequenzieren (Teilbereich von 2), Debugging (3) und Mustererkennung (Teilbereich von 4) zu beobachten waren. Da die Stichprobe mit 46 Kindern für ein Experimental-Kontrollgruppen-Design sehr klein ist, sind diese Ergebnisse aber nur bedingt belastbar. Auch wird nicht berichtet, wie der Lernzuwachs gemessen wurde. Città et al. (2019) können des Weiteren zeigen, dass das CT auch ohne den Einsatz digitaler Werkzeuge gefördert werden kann, nämlich dann, wenn die Kinder selbst in die Rolle eines Roboters schlüpfen und von anderen Kindern „programmiert“ werden. In dieser Studie mit Erst- bis Fünftklässler\*innen konnte außerdem ein positiver Zusammenhang zwischen mentaler Rotation und CT nachgewiesen werden. Dass auch die Förderung geometrischer Begriffe (Rechteck, Quadrat) mit dem Bee-Bot möglich ist, zeigt eine Langzeitstudie in einer ersten Klasse von Bartolini Bussi und Baccaglini-Frank (2015). Hierbei gelingt es auch schon sehr jungen Kindern Fahrtwege zu planen und ikonisch über Pfeilfolgen darzustellen.

## **Methodik**

In der im Folgenden vorgestellten Studie wurde eine Lernumgebung mit z. T. komplexen Problemaufgaben für Grundschulkinder ab Klasse 3 zum Bodenroboter Blue-Bot entwickelt. Ziel der Entwicklung ist es zu prüfen, ob sich beim Einsatz der Lernumgebung Lerngelegenheiten zu verschiedenen Teilkomponenten des CT zeigen. Gemäß des Design-Based-Research-Ansatzes (Prediger et al., 2012) wurden folgende Prototypen entwickelt: Einführung (Kennenlernen des digitalen Werkzeugs), Fahrtwege unterschiedlich darstellen (zur Vorbereitung der weiteren Module), Module zu den Teilkomponenten 2 bis 4 (Abb. 1). Die Prototypen wurden mit vier Tandems in

einem vierten Schuljahr in vier einstündigen Sitzungen erprobt. Die Einführung und die unterschiedliche Darstellung der Fahrwege wurden an einem Termin zusammengefasst. Die Videoaufzeichnungen wurden transkribiert. Die Auswertung erfolgte mittels der strukturierenden qualitativen Inhaltsanalyse (Mayring, 2015). Der Kodierleitfaden umfasst die Teilkomponenten 2 bis 4 als Oberkategorien. Die Subkategorien greifen Teiltätigkeiten innerhalb einer Oberkategorie auf, z. B. gehören zur Oberkategorie Debugging die Subkategorien Fehler finden und Fehler beheben.

Hier seht ihr einen Weg im kleinen Plan und eine Befehlsfolge. In der Befehlsfolge haben sich Fehler versteckt.

↑ ↶ ↑ ↑ ↶ ↑ ↑ ↷ ↓ ↓

1. Findet die Fehler und verbessert die Befehlsfolge.
2. Überprüft die verbesserte Befehlsfolge mit dem BlueBot.

Abb.1: Aufgabenbeispiel aus dem Modul Debugging (erstellt mit geogebra.org)

Die Auswertung mündete in das Re-Design der Aufgaben, welche erneut erprobt und weiterentwickelt wurden.

## Ergebnisse

Die Auswertung zeigte, dass bei der Bearbeitung der Prototypen Lerngelegenheiten zu den intendierten Teilkomponenten 2 bis 4 des CT beobachtet werden konnten. Es zeigte sich aber auch, dass die Teilkomponente Debugging in allen Modulen auftrat, auch im Einführungsmodul und im Modul „Fahrwege unterschiedlich darstellen“. Das Fehlerfinden, das Fehlerbeheben und das Evaluieren von Lösungen wurden in allen Modulen benötigt und von den Kindern auch durchgeführt. Fehler wurden von den Kindern immer dann erkannt, wenn sie einen Fahrweg in den Blue-Bot einprogrammierten

und den Blue-Bot anschließend diesen Weg durch Betätigung der Go-Taste auch abfahren ließen. Durch die ikonische Darstellung im Punkteraster konnten Fehler gut identifiziert werden. Das Generalisieren von Mustern und das Sequenzieren wurde hingegen nur in den dafür konzipierten Modulen beobachtet. Für die Gestaltung der Lernumgebung bedeutet dies, dass die Teilkomponente Debugging immanenter Bestandteil von Aufgabenstellungen zu Fahrtwegen im Gitterraster ist, während die Teilkomponenten Sequenzieren und Generalisieren durch spezifische Aufgabenstellungen ange-regt werden müssen.

## Literatur

- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J. & Zagami, J. (2016). A k-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, 19(3), 47–57.
- Bartolini Bussi, M. G. & Baccaglini-Frank, A. (2015). Geometry in early years: sowing seeds for a mathematical definition of squares and rectangles. *ZDM*, 47(3), 391–405.
- Caballero-Gonzalez, Y., García-Valcárcel, A. & García-Holgado, A. (2019). Learning computational thinking and social skills development in young children through problem solving with educational robotics. *TEEM '19: Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 19–23. <https://doi.org/10.1145/3362789.3362874>
- Città, G., Gentile, M., Allegra, M., Arrigo, M., Conti, D., Ottaviano, S., Reale, F. & Sciortino, M. (2019). The effects of mental rotation on computational thinking. *Computers & Education*, 141, 1-11.
- Eickelmann, B., Bos, W. & Labusch, A. (2019). Die Studie ICILS 2018 im Überblick – Zentrale Ergebnisse und mögliche Entwicklungsperspektiven. In B. Eickelmann, W. Bos, J. Gerick, F. Goldhammer, H. Schaumburg, K. Schwippert, M. Senkbeil, J. Vahrenhold (Hrsg.). *ICILS 2018#Deutschland. Computer- und informationsbezogene Kompetenzen von Schülerinnen und Schülern im zweiten internationalen Vergleich und Kompetenzen im Bereich Computational Thinking*. (7-32) Münster, New York: Waxmann.
- KMK- Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland (2017). *Strategie „Bildung in der digitalen Welt“*. Beschluss vom 8.12.2016, Stand: 09.11.2017. Online unter: [https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2017/Digitalstrategie\\_KMK\\_Weiterbildung.pdf](https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2017/Digitalstrategie_KMK_Weiterbildung.pdf) [10.12.2020]
- Mayring, P. (2015). *Qualitative Inhaltsanalyse. Grundlagen und Techniken* (12., überarb. Aufl.). Weinheim: Beltz.
- National Research Council. (2010). *Committee for the workshops on computational thinking: Report of a workshop on the scope and nature of computational thinking*. National Academies Press. <https://doi.org/10.17226/12840>
- Prediger, S., Link, M., Hinz, R., Hußmann, S., Thiele, J. & Ralle, B. (2012). Lehr-Lernprozesse initiieren und erforschen – Fachdidaktische Entwicklungsforschung im Dortmunder Modell. *MNU*, 65(8), 452–457.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.