

STOPPEL, Hannes & HÜSING, Sven  
Paderborn

## **Konstruktionistisches Geometrielernen durch epistemisches Programmieren in Scratch**

### **Einleitung**

In den letzten Jahren ist die Bedeutung von Computational Thinking (CT), nicht nur für den Informatikunterricht, sondern in verschiedenen Fächern immer relevanter geworden (Payne, Tawfik & Olney, 2022). Besonders hinsichtlich eines - ggf. auch fächerverbindenden - MINT-Unterrichts stellt CT eine erstrebenswerte Kompetenz dar, auch da es Schnittstellen mit den jeweiligen Fächern gibt (ebd.) Für die Mathematik bieten sich Anknüpfungspunkte - insbesondere hinsichtlich Problemlöseaktivitäten, die Parallelen zum Problemlösen in der Informatik aufweisen (Büscher, 2024).

In diesem Beitrag wollen wir anhand der Aktivität des epistemischen Programmierens (EP, siehe Hüsing et al., 2023) erkunden, wie CT dazu beitragen kann, mathematische Problemlöseaufgaben im Unterricht zu thematisieren und auf konstruktionistische Weise (siehe Papert & Harel, 1991) mathematische Erkenntnisse zu erlangen. Dazu wollen wir hier die Idee eines erkenntnisorientierten Programmierbegriffs für die Schule aufgreifen, der durch Verwendung von vorbereiteten Lernumgebungen in der Mathematik Anwendung finden kann, um Lernende mathematische Problemlöseprozesse im Unterricht angehen lassen zu können. Ziele des EP liegen in der Untersuchung des Erwerbs neuer Erkenntnisse im Rahmen iterativer Anpassungen des Programmcodes (Hüsing et al., 2023).

In Studien zeigten sich Verbindungen zwischen Beliefs (Überzeugungen) von Schüler\*innen in Bezug auf Programmierung und CT (Lee et al., 2024). Um Effekte des EP zu messen, bietet es sich an, Beliefs und ihre Veränderungen in Bezug auf Mathematik und Informatik zu untersuchen. Entsprechendes wird mithilfe von Prä- und Posttest mit Fragebögen durchgeführt.

Bzgl. der Studie möchten wir teilweise erprobtes Unterrichtsmaterial zum Thema Geometrie in Verknüpfung zu Arithmetik/Algebra in der Klasse 6 nach MSB NRW (2019) vorstellen, welches wir in Zukunft pilotieren und evaluieren.

### **Programmieren mit dem Ziel des Erkenntnisgewinns**

In vielen Disziplinen ist Programmieren heutzutage bereits ein fester Bestandteil im Kontext von Wissens- und Erkenntnisprozessen (Weintrop et al., 2016). Für den Mathematikunterricht bieten sich hier ebenfalls

In: L. Schick, M. Platz & A. Lambert (Hrsg.),  
Beiträge zum Mathematikunterricht 2025.

58. Jahrestagung der Gesellschaft für Didaktik der Mathematik. WTM.  
<https://doi.org/10.37626/GA9783959873307.0>

Anknüpfungspunkte, indem mathematische Konzepte durch das Programmieren eines Produktes, wie beispielsweise eines Spiels, vermittelt werden (Büscher, 2024; Misfeldt & Ejsing-Duun, 2015). Gerade im Bereich der Mathematik können Lernende durch Programmieren mathematische Sachverhalte genauer erkunden (siehe zum Beispiel Büscher, 2024).

In diesem Beitrag wollen wir diesbezüglich Ideen erprobten Unterrichts vorstellen, in denen die Methode des Programmierens im Mathematikunterricht dazu genutzt werden kann, im Sinne eines konstruktionistischen Ansatzes Erkenntnisse über mathematische Phänomene zu gewinnen. Ziel ist es dabei, dass Lernende der Klasse 6 durch Konstruktion geometrischer Objekte durch Verschiebung, Drehung, Spiegelung in einer visuellen Blockprogrammiersprache Erkenntnisse über die geometrischen Sachverhalte Polygon, Symmetrie und Rotation gewinnen und eine Einführung in CT und EP erfahren können.

### **Das Unterrichtsmaterial**

In Anbetracht der Tatsache, dass die Studie in Verbindung mit geometrischen Zusammenhängen der Lerninhalte der Stufe 6 in NRW durchgeführt werden soll, wurde auf die Programmierumgebung Scratch zurückgegriffen. Diese Wahl liegt unter anderem darin begründet, dass Scratch in den Kernlehrplänen und von der SWK (2022) explizit genannt wird und in zahlreichen Unterrichtsmaterialien der Stufen 5 und 6 für Informatik die häufigste Anwendung findet (vgl. Donderer et al., 2021; Hilbig et al., 2021).

Im Rahmen der Studie wurden von den Autoren Unterrichtssequenzen zu den Themen Polygon, Symmetrie, Rotation der Geometrie nach den Kernlehrplänen der Klasse 6 in NRW mit den inhaltlichen Schwerpunkten "Ebene Figuren", "Lagebeziehung und Symmetrie", "Abbildungen" der Geometrie und "Größen und Einheiten" der Arithmetik/Algebra (s. MSB NRW, 2019) unter der Anwendung von Scratch zur Einführung in CT und EP entwickelt und fanden erste Anwendung. Bei der Bearbeitung des Materials ist auf Kommentare innerhalb von Scratch sowie Lösungen im Heft zurückzugreifen. Abbildung 1 zeigt ein Beispiel zum Thema des Umfangs eines  $n$ -Ecks hin zur Zahl  $\pi$ .

### **Forschungsperspektive**

Ein Kern der Studie liegt in der Betrachtung des verwendeten Programmieransatzes mit dem Fokus auf den Gewinn von Erkenntnissen EP in Verbindung mit dem Mathematikunterricht. Dabei soll (a) der Zugang untersucht werden, den Schüler\*innen in Bezug auf die Betrachtung mathematischer Zusammenhänge wählen (mathematik- oder programmiergetrieben,

theoretisch oder explorativ,...), (b) wie sie Verknüpfungen zwischen den Fachbereichen ziehen, und (c) inwieweit geometrische Zusammenhänge oder auch Bezüge zur Arithmetik untersucht werden.

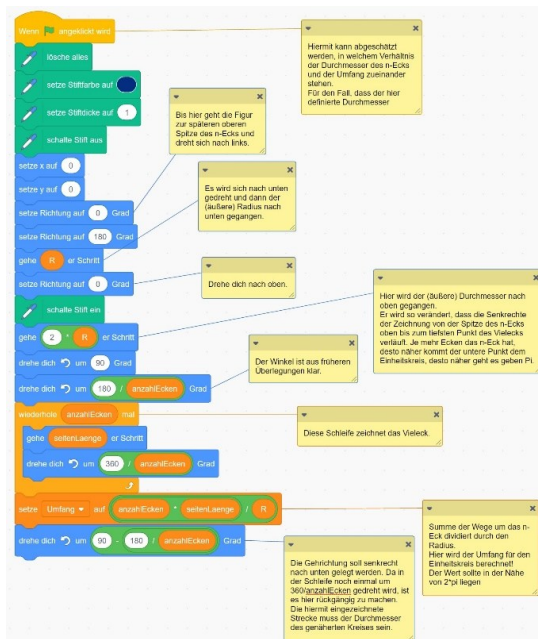


Abb. 1: Programm mit Scratch zum Umfang eines  $n$ -Ecks und  $\pi$

Der Untersuchung dienen die Scratch-Programme inklusive der Kommentare im Programm sowie die Dokumentation der Schüler\*innen in ihrem Heft. Die Programme sind als Möglichkeit der Notiz von Erkenntnissen im Bereich der Informatik gedacht. Das Heft erfüllt den Zweck der Untersuchung des Verständnisses der mathematischen Lerninhalte und damit des Transfers der Ergebnisse des Programmierens in die Mathematik und umgekehrt. Zur Analyse dieses Transfers wurde ein Schema für die Analyse von Zusammenhängen zwischen Verständnis der Mathematik und Programmieren entwickelt, mit dem sich Verknüpfungen zwischen Erkenntnissen in EP & CT und dem Verständnis von Strukturen der mathematischen Zusammenhänge inklusive der Brücke zwischen Geometrie und Algebra ziehen lassen.

Im Zusammenhang mit EP sollen Erfolge von Schüler\*innen und ihre Beliefs im Fachlichen untersucht werden, um Verknüpfungen zwischen Ansatz einer Lösung und Beliefs von Schüler\*innen sichtbar zu machen, so dass sich Wege finden lassen, Motivation und Kompetenzen von Schüler\*innen in Mathematik und Informatik zu erhöhen. Zur Analyse solcher Effekte wurden auf Basis der Studien von Kloosterman & Stage (1992), Lee et al. (2024) und nach Erkenntnissen von Stoppel (2019) Fragebögen mit einer fünfteiligen Likert-Skala zu Beliefs bzgl. Mathematik, Informatik in Bezug auf EP inklusive Verknüpfungen zwischen den Komponenten für Pre- und Post-Test entwickelt, so dass sich Beliefs der Schüler\*innen u.a. in Bezug auf die

"Bedeutung des Programmierens für die Mathematik" und ihre "Fähigkeit der algorithmischen Lösung mathematischer Probleme" erfassen lassen, um Verbindungen zwischen Programmierung, Kommentaren und mathematischen Erkenntnissen zu erfassen und zu analysieren.

## Literatur

- Büscher, C. (2024). Differences in Students' Computational Thinking Activities when Designing an Algorithm for Drawing Plane Figures. *International Journal of Science and Mathematics Education*. <https://doi.org/10.1007/s10763-024-10465-3>
- Donderer, M., et al. (2021). *Informatik 5/6*. Berlin: Cornelsen.
- Hilbig, A., et al. (2021). *Starke Seiten 5/6*. Stuttgart: Klett.
- Hüsing, S., Schulte, C., & Winkelkemper, F. (2023). Epistemic Programming. In S. Sentance, E. Barendsen, N. R. Howard, & C. Schulte (Hrsg.), *Computer Science Education: Perspectives on Teaching and Learning in School* (2. Aufl., S. 291–304). Bloomsbury Academic; Bloomsbury Collections. <http://dx.doi.org/10.5040/9781350296947.ch-022>
- Kloosterman, Peter; Stage, F.K (1992): Measuring beliefs about mathematical problem solving. In: *School science and mathematics* 92 (3), S. 109–115.
- Lee, S. W.-Y., Liang, J.-C., Hsu, C.-Y., & Tsai, M.-J. (2024). Students' beliefs about computer programming predict their computational thinking and computer programming self-efficacy. *Interactive Learning Environments*, 32(8), 4088–4108.
- Misfeldt, M., & Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. In K. Krainer & N. Vondrová (Hrsg.), *CERME 9—Ninth congress of the european society for research in mathematics education* (S. 2524–2530). Charles University in Prague, Faculty of Education and ERME. <https://hal.science/hal-01289367>
- MSB NRW (2019). Kernlehrplan für die Sekundarstufe I in NRW - Mathematik. <https://www.schulentwicklung.nrw.de/lehrplaene>
- Papert, S., & Harel, I. (1991). Situating Constructionism. In S. Papert & I. Harel (Hrsg.), *Constructionism* (S. 1–12). Ablex Publishing Corporation. <http://www.papert.org/articles/SituatingConstructionism.html>
- Payne, L., Tawfik, A., & Olney, A. M. (2022). Computational Thinking in Education: Past and Present. *TechTrends*, 66(5), 745–747. <https://doi.org/10.1007/s11528-022-00766-1>
- Stoppel (2019). *Beliefs und selbstreguliertes Lernen*. Springer Fachmedien Wiesbaden.
- SWK (2022). *Digitalisierung im Bildungssystem: Handlungsempfehlungen von der Kita bis zur Hochschule*. [https://www.swk-bildung.org/content/uploads/2024/02/SWK-2022-Gutachten\\_Digitalisierung.pdf](https://www.swk-bildung.org/content/uploads/2024/02/SWK-2022-Gutachten_Digitalisierung.pdf)
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>