

Mathematische Vorstellungen entwickeln durch Programmieren

1. Einführung

In der Mathematikdidaktik gibt es eine lange Tradition, in der in konstruktivistischen Lehr-/Lernszenarien die Entwicklung von mathematischen Vorstellungen / mentalen Modellen durch eigenes Programmieren in kindgerechten Sprachen wie Logo oder Scratch gefördert werden soll.

Im Rahmen des von der Telekom Stiftung geförderten Projekts „Digitales Lernen in der Grundschule Stuttgart/Ludwigsburg (dileg-SL)“ wurden an der Pädagogischen Hochschule Ludwigsburg in einem Seminar solche Unterrichtseinheiten für Mathematik entwickelt und von den Studierenden in einer Grundschule in Stuttgart erprobt. Dieses Projektseminar wird wissenschaftlich begleitet. Dazu finden Vorher-Nachher-Erhebungen der Computernutzerselbstwirksamkeitserwartung, eine Analyse der Programmierprozesse mit Hilfe von Screenvideos sowie eine Abschlussdiskussion statt.

2. Theoretischer Hintergrund

Die Idee, dass über das eigene Programmieren – oder inzwischen eher „Coding“ genannt – Lernende mathematische Vorstellungen entwickeln können, geht auf die 1980er Jahre und die Gruppe um Seymour Papert (1980) zurück. Durch das direkte Feedback des Computers können sich Schülerinnen und Schüler selbst immer wieder klar machen, ob ihre Vorstellung zu der „Realität“ passt. Inzwischen mischt sich in diesem Kontext das Konzept des "Computational Thinking". Wing (2006) benennt unter anderem das Konzeptualisieren sowie das Denken auf verschiedenen Abstraktionsebenen als wichtige Aspekte des Computational Thinking. Für sie ist dies eine menschliche und keine maschinelle Art zu denken, und zwar eine Kombination aus mathematischem und technischem Denken.

Zu Lehr-/Lernszenarien im Kontext Computational Thinking gibt es verschiedene didaktische Konzepte. Da ist zum einen der Aktivitätsrahmen „5 E“ von Benton, Hoyles, Kalas & Noss (2017), der für das Projekt ScratchMath entwickelt wurde. Hier betonen die Autoren, dass die Aktivitäten: Explorieren/Untersuchen, Vorstellen/Voraussagen, Erläutern/Erklären, Austausch/Kommunikation, Verbinden/Anwenden in erfolgreichen Lehr-/Lernszenarien enthalten sein müssen, damit das Ziel – die Entwicklung mathematischer Vorstellungen – erreicht werden kann. Eine andere Gruppe

(Kotsopoulos, Floyd, Khan, Namukasa, Somanath, Weber & Yiu, 2017) beschreibt einen Pädagogischen Rahmen, der die Bereiche Unplugged, Tinkering, Making und Remixing umfasst. Wichtig ist hierbei, dass immer mit dem „Unplugged-Bereich“ angefangen wird.

3. Konzeption der Veranstaltung und der Evaluation

In der Veranstaltung entwickelten die insgesamt 17 Studierenden (alle Lehramt, 15 mit Mathematik als Fach), nachdem sie erste Erfahrungen mit dem Programmieren mit Scratch gemacht und die oben genannten didaktischen Szenarien kennengelernt hatten, in Zweiergruppen Lernumgebungen für Schülerinnen und Schüler einer 3. Klasse zu mathematischen Themen. Dabei waren eine „unplugged-Einführung“ in das mathematische Thema und Phasen eigenen Programmierens der Kinder verpflichtend vorgegeben. Dazu wurde ein Planungsraster entwickelt, das alle nutzen sollten.

Die Evaluation erfolgte über eine Vorher-Nachher-Befragung zur Computernutzerselbstwirksamkeit CUSE (Spannagel & Bescherer, 2009) sowie zur Selbstwirksamkeit in Bezug auf den Einsatz von Computern im Unterricht SECU (Dinse da Salas, Rohlf's & Spannagel, 2016) der Studierenden, die Analyse der Lernumgebungen sowie Screenvideos der Programmierphasen der Kinder und eine Abschlussdiskussion mit den Studierenden. In dem Gesamtprojekt dileg-SL wurden noch weitere Evaluationsinstrumente wie ein Projektjournal und ein medienpädagogischer Schülerfragebogen) eingesetzt, die hier aber nicht weiter erörtert werden.

4. Ergebnisse der Evaluation

An der Veranstaltung nahmen 17 Studierende (14 weiblich, 3 männlich) teil, 14 Personen haben sowohl die Vor- wie auch die Nachherbefragung ausgefüllt. Es entstanden 8 ausgearbeitet Lernumgebungen zu den mathematischen Themen: kleinstes gemeinsames Vielfaches, Simulation von Ampelschaltungen, Weg der Müllabfuhr (Eulerkreis), Orientierung auf der Ebene (Himmelsrichtungen/Koordinatensystem/rechts, links), Kombinatorik, Muster aus Quadraten erstellen sowie Talervervielfältigung (Funktionsmaschine).

Die Studierenden hatten jedoch in ihren Planungsrastern große Schwierigkeiten, die mathematischen Grundlagen ihrer Lernumgebungen darzustellen.

Dagegen wandten sie viel Mühe für die Gestaltung des „unplugged-Einstiegs“ und der grafischen Oberfläche der Lernumgebungen auf.

In den Befragungen zur Selbstwirksamkeitserwartung konnte statistisch kein Unterschied zwischen den Mittelwerten der Vorher- und Nachher-Ergeb-

nisse festgestellt werden. Bei Betrachtung der Mittelwerte nahm die Computernutzerselbstwirksamkeit sogar leicht ab (Mittelwert $CUSE_{pre}$ 57,00 / Mittelwert $CUSE_{post}$ 55,46)¹.

Da die Situation, in der die 17 Studierenden mit 16 Schülerinnen und Schüler die Lernumgebungen erprobt hatten, sehr weit von einem normalen Mathematikunterricht entfernt waren, ist der gleichbleibende Wert des SECU-Mittelwerts nicht überraschend. Enttäuschend dagegen ist der Abfall der Mittelwerte der drei Fragen im CUSE-Fragebogen, die sich auf das Lernen mit Computern beziehen:

- Item 5: Das Verwenden von Computern macht Lernen interessanter.
Mittelwert_{pre} = 4,64 / Mittelwert_{post} = 3,86²
- Item 8: Computer sind gute Hilfsmittel beim Lernen.
Mittelwert_{pre} = 4,86 / Mittelwert_{post} = 4,38
- Item 10: Einige Computerprogramme machen Lernen eindeutig einfacher.
Mittelwert_{pre} = 5,29 / Mittelwert_{post} = 4,43

Die Analyse der Screenvideos zeigt, dass zwar immer wieder das für das Computational Thinking typische „Testen“ und „Debugging“ zu erkennen sind. Beides bezieht sich jedoch v.a. auf den Umgang mit Scratch an sich (e beim Ziehen und Zusammenklicken der Blöcke sowie Eingeben von Parametern). Eine systematische Überprüfung der mathematischen Vorstellungen und entsprechende Anpassung des Programmes sind praktisch nicht erkennbar.

In der Reflexionsrunde wurde für die schwache Auseinandersetzung mit den mathematischen Inhalten – sowohl der Studierenden wie auch der Schülerinnen und Schüler – v.a. der hohe Zeitaufwand für die Einarbeitung in Scratch an sich und auch die Faszination mit den Möglichkeiten von Scratch genannt. Als sehr positiv bewerteten die Studierenden den „unplugged-Einstieg“ sowie die Möglichkeit ihre Lernumgebung ganz frei zu wählen und „was eigenes zu machen“.

5. Konsequenzen für die Veranstaltung

Im folgenden Semester (Sommer 2018) wird die Veranstaltung im Rahmen des Projekts dileg-SL noch einmal angeboten und es wurden einige Änderungen aufgrund der Erfahrungen aus dem Vorsemester beschlossen. Obwohl die Studierenden die freie Wahl des Themas ihrer Lernumgebung so

¹ Fragebogen mit 12 Items, 6-teilig likertskaliert mit Werten 1 bis 6.

² Min 1 und Max 6

schätzten, werden die mathematischen Themen auf die „Orientierung in der Ebene“ und „Muster mit ebenen, regelmäßigen Figuren“ eingeschränkt. Dies soll eine Fokussierung auf die mathematischen Vorstellungen zu diesen beiden Themenbereichen erleichtern.

Die Selbsterfahrungsaufgabe, bei der die Studierenden eigene Erfahrungen beim Mathematiklernen durch Programmieren machen, wird inhaltlich reduziert und das Mathematik lernen stärker in den Vordergrund gestellt.

Damit die didaktischen Konzepte, v.a. die verschiedenen Aktivitäten des „E 5“-Modelles, auch tatsächlich genutzt werden, müssen alle fünf sowohl in den Planungsrastern wie auch in den Lernumgebungen vorkommen. Um die Erprobungsphase in der Schule noch besser didaktisch reflektieren zu können wird geplant, zwischen den beiden Besuchen in der Schulen eine Seminarsitzung an der Hochschule einzufügen.

Insgesamt sollen jedoch auf jeden Fall die sehr hohe Motivation und das Engagement der Studierenden sowie der Schülerinnen und Schüler beim Thema mathematische Vorstellungen entwickeln durch Programmieren beibehalten werden.

Das Projekt dileg-SL wird von der Deutschen Telekom Stiftung gefördert.

Literatur

- Benton, L., Hoyles, C., Kalas, I. & Noss, R. (2017). Bridging Primary Programming and Mathematics: some findings of design research in England. *Digital Experiences in Mathematics Education*, 3, S. 115-138.
- Dinse de Salas, S., Rohlfs, C. & Spannagel, C. (2016). Coaching teachers in using technology. In *Proceedings of EdMedia 2016 – World Conference on Educational Media and Technology* Vancouver, BC, Canada: Association for the Advancement of Computing in Education (AACE). (S. 927-934). Online unter <https://www.learn-tech-lib.org/p/173060/>, Zugriffsdatum 10.04.2018.
- Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, 3, S. 154-171.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Spannagel, C. & Bescherer, C. (2009). Computerbezogene Selbstwirksamkeitserwartung in Lehrveranstaltungen mit Computernutzung. *Notes on Educational Informatics, Section A: Concepts and Techniques*, 5(1), S. 23-43.
- Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), S. 33-35.