

SCHÜRMAN, Uwe; BRUCKMAIER, Georg & REPENNING, Alexander
MuttENZ, Brugg-Windisch

Game-Design als Zugang zum mathematischem Problemlösen

Abstract

Computational Thinking (CT) kann nicht nur im Informatik-, sondern auch im Mathematikunterricht dazu dienen, fachliche Inhalte zu fördern. Aufbauend auf Paperts konstruktionistischem Ansatz untersucht dieser Beitrag, welche Formen mathematischen Problemlösens von Lernenden der sechsten Jahrgangsstufe sich beim Programmieren von Spielen mit Zufallselementen zeigen. Vier Schülerpaare aus zwei Klassen gestalteten in der visuellen Programmierumgebung RULER.game ein Whac-a-Mole-Game. Die Analyse von Videodaten und Stimulated-Recall-Interviews zeigt: Der Design-Prozess folgt iterativen, zyklischen Mustern, in denen Phasen wie das Planen, Implementieren, Testen und Evaluieren mehrfach durchlaufen werden.

Einleitung und theoretische Rahmung

CT umfasst die Fähigkeit, Probleme systematisch zu analysieren, in Teilprobleme zu zerlegen und algorithmisch zu lösen (Wing, 2006). Die Ursprünge von CT gehen auf Paperts *Konstruktionismus* zurück, dessen *Power Principle* ein Lernen betont, das durch aktive Gestaltung digitaler Artefakte wie z. B. Videos, Dateien oder Codes zum Begriffsverständnis beiträgt – anstatt eher rezeptiv und passiv Konzepte zu lernen. In dieser Tradition wird Programmieren im Sinne des *Programming to Learn* nicht als technisches Ziel, sondern als epistemisches Werkzeug verstanden (Papert, 1980, 1986).

Game-Design, bei dem Lernende digitale Artefakte nicht nur selbst programmieren, sondern auch kreativ gestalten, bietet die Möglichkeit, den Mathematikunterricht mit CT im Sinne des *mathematischen Problemlösens* (Schoenfeld, 1992) zu Computational Math Thinking (CMT) zu verbinden. So teilen Programmierprozesse zentrale Merkmale mit mathematischen Problemlöseprozessen wie etwa Problemdekomposition, Abstraktion und Ergebnisevaluation (Shute et al., 2017).

Fragestellung und Methoden

Die vorliegende explorativ-deskriptive Studie untersucht die folgende Forschungsfrage (FF): Wie gestaltet sich der Game-Design-Prozess als Problemlöseprozess beim Programmieren von Spielen mit Zufallselementen bei Lernenden der Jahrgangsstufe 6? Insbesondere:

- FF1: Wie häufig und in welcher Reihenfolge durchlaufen Lernende Teilprozesse eines Programmier- bzw. Problemlöseprozesses (z. B. Planen, Implementieren, Testen und Evaluieren)?
- FF2: Wie gehen Lernende mit Herausforderungen und Fehlern im Prozess um?

Vier leistungshomogene Schülerpaare aus der sechsten Jahrgangsstufe zweier Schweizer Primarschulen arbeiteten in der visuellen Programmierumgebung RULER.game an einem Whac-a-Mole-Game, einem Computerspiel, bei dem zufällig erscheinende Maulwürfe angeklickt werden sollen. Dabei galt es, Wahrscheinlichkeit und Frequenz der auftauchenden Maulwürfe so zu programmieren, dass das Spiel herausfordernd, aber nicht zu schwer ist. Datengrundlage für die Auswertung bildeten Videografien des Game-Design-Prozesses sowie Stimulated-Recall-Interviews mit jeweils einem Mitglied pro Schülerpaar, die anschließend in einer *qualitativen Inhaltsanalyse* (Mayring, 2010) ausgewertet worden sind.

Zur Gewinnung informationsreicher Fälle wurde eine gezielte Stichprobenauswahl vorgenommen (Nyimbili & Nyimbili, 2024). Dabei wurden zwei Sampling-Strategien kombiniert: Homogenität innerhalb der Paare (Lernende derselben Jahrgangsstufe mit demselben Mathematiklehrwerk) sowie maximale Variation zwischen den Paaren (Leistungsunterschiede und Programmiererfahrung). Als deduktive Kategorien für die Kodierung dienten typische Prozessschritte des Programmierens bzw. Problemlösens (FF1).

Daneben erfolgte eine induktive Kategorienbildung durch Subsumtion (Schreier, 2014), um etwa den Umgang mit Fehlern und Herausforderungen (FF2) zu erfassen. So ergab sich eine Unterscheidung hinsichtlich der Kommunikation (Einzel- oder Zwiegespräch, Aussage mit oder ohne Begründung). Ferner wurden in der Auswertung auch Ausprägungen von Kategorien (z. B. Häufigkeit und zeitliche Gewichtung einzelner Problemlöseprozessschritte) quantitativ gefasst. Somit wurde das Verfahren der strukturierenden qualitativen Inhaltsanalyse durch Aspekte der evaluativen bzw. skalierenden qualitativen Inhaltsanalyse ergänzt (Mayring, 2010; Mayring & Fenzl, 2022).

Ergebnisse

Alle Schülerpaare durchliefen bei der Aufgabenbearbeitung iterativ mehrere Zyklen, bestehend aus Planen, Implementieren, Testen und Evaluieren (vgl. Abb. 1). Dabei verbrachten gerade die leistungsstarken Paare (Schülerpaare 1 und 3) relativ viel Zeit mit dem Testen im Vergleich zum Implementieren.

Die Analyse macht deutlich: Die Prozesse verlaufen nicht linear-top-down, sondern ähneln eher einem iterativen Tüfteln (Resnick & Rosenbaum, 2013).

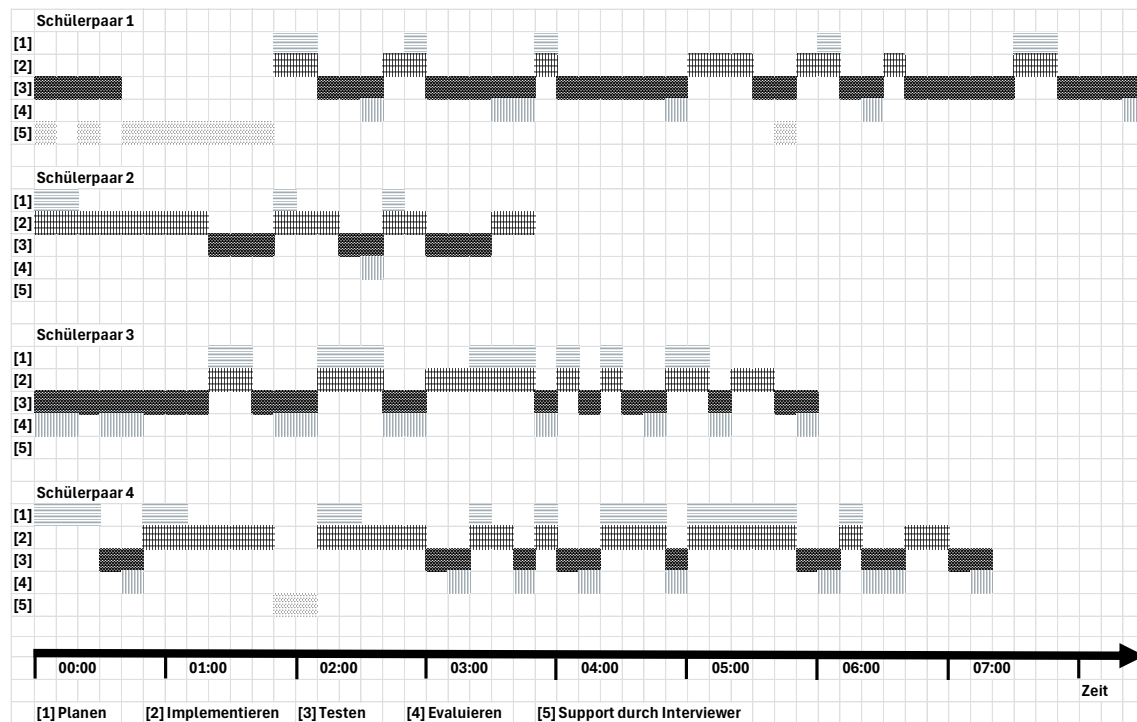


Abb. 1: Bearbeitungszyklen der Schülerpaare

Bezüglich des Umgangs mit Herausforderungen im Prozess wurde zudem die Kommunikation der Schülerpaare untersucht. Die leistungsstarken Paare, also die Schülerpaare 1 und 3, verbrachten mehr Zeit mit verbaler Kommunikation (Planen und Evaluieren). Das leistungsstärkste Schülerpaar 3 zeigte eine besonders dichte, qualifizierte Kommunikation, in der Hypothesen gebildet und durch Testen überprüft wurden. Im Gegensatz zur Mathematikleistung und dem Vorwissen zu Wahrscheinlichkeiten hatte die Programmiererfahrung der Schülerpaare keinen erkennbaren Einfluss auf die Performanz während des Game-Design-Prozesses.

Diskussion und Ausblick

Ziel der vorliegenden Studie war es, das Zusammenspiel zwischen mathematischem Problemlösen und Aspekten von CT herauszuarbeiten. Die Ergebnisse liefern erste Hinweise darauf, wie Game-Design als didaktisches Szenario zur Förderung des mathematischen Problemlösens im Sinne von CMT genutzt werden kann. So formulierten die Lernenden Hypothesen, testeten diese und passten das von ihnen erstellte Spiel iterativ an. Im Mathematikunterricht könnten derlei CT-Praktiken zur Förderung des Problemlösens (und des probabilistischen Denkens) gezielt angebahnt werden, etwa

durch explizite Planungsphasen, gezielten Debugging-Aufgaben oder Simulationen mit variablen Spielparametern.

Bedingt durch das explorative Design, die kleine Fallzahl und die Kontextgebundenheit mit nur einer Aufgabenstellung erlaubt die Untersuchung noch keine generalisierbaren Aussagen. Um zu solchen Schlussfolgerungen zu gelangen, sollten zukünftige Studien empirisch untersuchen, wie sich das mathematische Problemlösen (und das probabilistische Denken) mittels Game-Design über *längere Zeiträume* entwickelt, welche *Unterstützungsmaßnahmen* die Begriffsbildung gezielt fördern können und inwiefern sich solche Lernsettings auch auf andere *mathematische Prozess- und Inhaltskompetenzen* übertragen lassen. Um in diesem Kontext Wirkzusammenhänge objektiv erfassen zu können, wäre zudem eine Ergänzung um quantitative Methoden wünschenswert.

Literatur

- Mayring, P. (2010). *Qualitative Inhaltsanalyse—Grundlagen und Techniken* (11. aktualisierte und überarbeitete Auflage). Beltz.
- Nyimbili, F., & Nyimbili, L. (2024). Types of Purposive Sampling Techniques with Their Examples and Application in Qualitative Research Studies. *British Journal of Multidisciplinary and Advanced Studies*, 5(1), 90–99. <https://doi.org/10.37745/bjmas.2022.0419>
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.
- Papert, S. (1986). *Constructionism: A New Opportunity for Elementary Science Education*. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group.
- Resnick, M., & Rosenbaum, E. (2013). Designing for Tinkerability. In M. Honey & D. Kanter (Hrsg.), *Design, Make, Play: Growing the Next Generation of STEM Innovators* (S. 163–181). Routledge.
- Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics. In D. Grouws (Hrsg.), *Handbook of research on mathematics teaching and learning: A project of the National Council of Teachers of Mathematics* (S. 334–370). Macmillan.
- Schreier, M. (2014). Varianten qualitativer Inhaltsanalyse: Ein Wegweiser im Dickicht der Begrifflichkeiten. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 15(1). <https://doi.org/10.17169/fqs-15.1.2043>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>