



Unterrichtsexperimente – Impulse für wirksames Lernen

Abstract einer Dokumentation	
Titel der Arbeit	Förderung von „Computational Thinking“ durch den Einsatz einer ikonischen Programmiersprache (Scratch) im Fach Informationstechnik in der Eingangsklasse des TGI unter besonderer Berücksichtigung ausgewählter Strukturelemente (Sequenz, Auswahl und Wiederholung) und Variablen.
Verfasser/in	J. Dominik Krauß
Fach	Informationstechnik
Ausgangslage - Problemstellung	<p>In der Eingangsklasse des Technischen Gymnasiums ist es im Fach Informationstechnik vorgesehen, den Lernenden die Grundlagen der Programmentwicklung zu vermitteln. Häufig lässt sich dabei in der vorherrschenden Lehre ein hoch abstraktes Vorgehen beobachten, in dem konzeptuelle Programmierfähigkeiten und syntaktisches Wissen über die eingesetzte Programmiersprache im Kollektiv vermittelt werden. Dieses Vorgehen führt gerade bei Lernenden mit wenig Vorerfahrung im Programmieren dazu, dass die syntaktisch korrekte Formulierung des Programmcodes in den Fokus rückt und die Vermittlung der theoretischen Konzepte in den Hintergrund tritt.</p> <p>Aufgrund der Austauschbarkeit einzelner Programmiersprachen ist bei diesem Vorgehen infrage zu stellen, ob der Fokus der aktiven Lernzeit bei der Einführung des Themas auf der Lösung syntaktischer Fehler liegen sollte oder ob der Vermittlung der konzeptuellen Fertigkeiten ein größerer Raum im Unterricht eingeräumt wird. Diesem alternativen Ansatz folgt das Konzept des Computational Thinking (CT).</p> <p>Der Begriff Computational Thinking kann auf Seymour Papert zurückgeführt werden und erlangte größere Bekanntheit durch Jeannette Wing, die in ihrer Arbeit Computational Thinking als eine grundlegende Fertigkeit beschreibt, die jedes Kind neben Lesen, Schreiben und Rechnen als analytische Fähigkeit erlernen sollte. Nach Wing verfolgt Computational Thinking dabei den Ansatz, durch fundamentale Konzepte der Informationstechnik Probleme zu lösen, Systeme zu designen oder das menschliche Verhalten zu verstehen (vgl. Wing 2006: 33ff. und Wing 2008: 3717ff.).</p> <p>Diesem Ansatz folgend, stellt die Dokumentationsarbeit eine Idee vor, die die Vermittlung dieser abstrakteren konzeptuellen Fähigkeiten in den Fokus des Unterrichts rückt.</p>
Darstellung des Unterrichtsexperiments – Ansatz zur Lösung des Problems	<p>Basierend auf dem Ansatz des Computational Thinking findet eine Umstrukturierung des unterrichtlichen Vorgehens statt. Die Erarbeitung und Vertiefung der theoretischen Konzepte wird vorgelagert und anhand einer intuitiv verständlichen ikonischen Programmiersprache (Scratch) in den Fokus gestellt. Durch den Einsatz dieser blockbasierten Programmiersprache wird es ermöglicht, sich von syntaktischen Problemstellungen zu lösen und den konzeptuellen Sinn und Mehrwert der einzelnen Artefakte zu beleuchten.</p> <p>Verzahnt wird dabei jeweils ein neues Strukturelement in der ikonischen Programmiersprache eingeführt und in den Folgestunden auf die eingesetzte textbasierte Programmiersprache übertragen. Anschließend wird zyklisch ein weiteres Strukturelement eingeführt.</p>



<p>Umsetzung und Reflexion: Was hat funktioniert? Was müsste anders gemacht werden?</p>	<p><u>Umsetzung:</u> Zur empirischen Betrachtung der Auswirkungen dieses experimentellen Unterrichtsversuchs wird ein kontrolliertes, aber nicht randomisiertes Interventionsstudiendesign nach dem Pre- und Posttest-Prinzip gewählt, d. h. die Ergebnisse der Studiengruppe, werden mit einer Kontrollgruppe (Referenzwert) verglichen. Dazu werden beide Gruppen sowohl vor als auch nach der Intervention (die nur die Studiengruppe erfuhr) getestet und deren Entwicklung verglichen. Als Kontrollgruppe dient eine Parallelklasse der Schule mit ähnlichen Ausgangsvoraussetzungen bezüglich der Gruppengröße und des erreichten Bildungsabschlusses.</p> <p>Um die Effekte der Intervention auf die CT-Fähigkeiten zu messen und einzuordnen, werden verschiedene Evaluationsinstrumente genutzt. Erstens werden Beobachtungen und Einschätzungen durch die Lehrpersonen berücksichtigt. Zweitens kommen Meinungsumfragen, in denen die SuS eine Selbsteinschätzung zu ihren Überzeugungen und Fähigkeiten vornehmen und Erfahrungen mit der Unterrichtseinheit rückmelden können, zum Einsatz. Drittens wird ein wissenschaftlich empirisch geprüfter und damit validierter Wissenstest verwendet, um den Entwicklungsstand der SuS in Bezug auf die CT-Fähigkeiten zu erfassen. Hierzu wird der durch Marcos Román-González entwickelte Computational-Thinking-Test (CT-Test) für die Entwicklung eines Tests zugrunde gelegt (siehe auch Román-González 2015), der weltweit in der CT-Forschung bei führenden Studien eingesetzt wird.</p> <p><u>Reflexion:</u> Aus Sicht der Lehrperson kann festgehalten werden, dass der Einsatz einer ikonischen Programmiersprache motivational positive Effekte hervorruft und die Lernenden trotz des noch nicht ausgeprägten Vorwissens in den Unterrichtsstunden mit dem Fokus auf ikonischer Programmierung bereits sehr selbstständig und aktiv arbeiten können. Probleme bei der Programmierung ergeben sich auffällig selten. Bei der Übertragung der Konzepte auf die textbasierte Sprache kann dennoch ein erwarteter Rückgang der Selbstständigkeit festgestellt werden, die dem erwarteten Maß aus dem klassischen Unterrichtsvorgehen entspricht. Auffällig zentral stellen sich in diesen Stunden Fragen zu syntaktischen Problemen dar. Probleme mit dem konzeptuellen Verständnis der Strukturen ergeben sich vergleichsweise selten, was auf ein fundiertes Verständnis der Konzepte deutet.</p> <p>Aus der Umfrage unter den Lernenden ist hervorzuheben, dass eine große Mehrheit rückmeldet, dass die Arbeit mit Scratch Freude bereitet und die SuS gerne mehr mit Scratch arbeiten würden. Außerdem empfinden die SuS mehrheitlich, dass ihnen die Arbeit mit Scratch den Einstieg in die Programmierung erleichtert.</p> <p>Auch die Ergebnisse aus dem empirischen Wissenstest zeigen eine positive Entwicklung des Unterrichtsarrangements auf die CT-Fähigkeiten der Lernenden. Zusammengefasst lässt sich zeigen, dass die Versuchsgruppe den Anteil korrekt beantworteter Fragen um deutliche 18 % steigern konnte. Die Kontrollgruppe, die keine Intervention erfuhr, zeigte mit einer Steigerung um 1 % wie erwartet ein nur leicht verändertes Ergebnis.</p>
---	---



<p>Konkrete Handlungsempfehlungen für den Unterricht</p>	<ul style="list-style-type: none"> - Computational Thinking stellt einen bedeutenden Kompetenzbereich dar, der es wert ist, explizit geschult zu werden. - Im Jahresplan sollten Zeiten geplant werden, innerhalb derer das konzeptuelle Verständnis der Strukturelemente explizit thematisiert wird. - Der vorgelagerte Einsatz einer ikonischen Programmiersprache bietet sich an, um die Motivation und Selbstwirksamkeit der Lernenden beim Erlernen des Programmierens zu steigern. - Der vorgelagerte Einsatz einer ikonischen Programmiersprache bietet sich an, um im Lernprozess den Fokus auf die Entwicklung konzeptioneller CT-Fähigkeiten zu legen.
--	--