

Klaus-Tycho FÖRSTER, Hildesheim

Neue Möglichkeiten durch die Programmiersprache Scratch: Algorithmen und Programmierung für alle Fächer

Das Institut für Mathematik und Angewandte Informatik (IMAI) ist an der Universität Hildesheim seit mehr als 10 Jahren zuständig für die Ausbildung in Informations- und Kommunikationstechnologie (IuK) für alle Studierenden des Lehramtes an Grund-, Haupt- und Realschulen. Wie in den meisten lehramtsausbildenden Studiengängen werden auch in Hildesheim in der IuK-Ausbildung grundlegende IT-Kompetenzen für den späteren schulischen Alltag bereitgestellt. Seit diesem Wintersemester vermitteln wir erstmals vertieft auch IT-Basiskompetenzen im Bereich Algorithmen und Programmierung – ermöglicht durch die visuelle Programmiersprache Scratch: nicht nur für unsere zukünftigen Mathematiklehrer, sondern auch für die Lehramtsstudierenden aller Fächer – also z.B. auch für die zukünftigen Grundschullehrer mit den Fächern Deutsch und Religion.

Algorithmen und Programmierung in Mathematik und Informatik

Schon 1981 forderte die GDM für die Ausbildung aller Mathematiklehrer, dass diese mindestens eine Programmiersprache beherrschen sollen und darin die Algorithmen ihres Unterrichts programmieren und realisieren können sollen (vergl. Weigand 1989). Die Bedeutung von Algorithmen für die mathematische oder informatische Ausbildung ist seit langem bekannt, siehe z.B. Tietze/Klika/Wolpers 1997, Modrow 2006 oder Schwill 1995.

Eine der zentralen Begründungen für das Programmieren im mathematischen Unterricht sind häufig auftretende Schwierigkeiten der Schüler beim Problemlösen. Programmieren kann hierbei helfen, da es eine gemeinsame Sprache bietet, eigene Erfahrungen ermöglicht und man über Programme leichter reden kann – insbesondere über ihre Struktur, ihre Entwicklung und ihre Beziehungen zu anderen Programmen (vergl. Feurzeig und Papert 1969). Programmieren ist aber auch hilfreich für eine Überprüfung und Bewertung komplexer Modellierungen und ist förderlich für die Kritikfähigkeit am (richtigen?) Ergebnis. So führten etwa schon Howe et. al. 1988 aus, dass Schüler beim Programmieren ausgehend vom gesehenen Ergebnis ihre Ideen wiederholt modifizieren. Ebenso finden sich bei der Programmierung Parallelen zum mathematischen Beweisen: *"Das Finden einer Transformation, die Eingabewerte auf Ausgabewerte abbildet, das Finden eines Algorithmus oder einer Konfiguration, kann man mit einer Beweisaufgabe im Mathematikunterricht vergleichen. [...] Dieser Vergleich ergibt sich aus den Arbeiten von Regina Bruder (Bruder, Leuders, Büchter*

2008)." (Strecker 2009, S. 47). Das Thema der Programmierung im mathematischen Unterricht ist uneingeschränkt aktuell (s. Kortenkamp 2008).

Trotz aller geschilderten Argumente ist im mathematischen Unterricht heutzutage das Programmieren aber eher auf dem Rückzug. Dieses ist unter anderem bedingt durch die Verbreitung von mathematischer Spezialsoftware, aber auch insbesondere vermutlich durch den zu hohen Aufwand des Erlernens einer (Standard-) Programmiersprache durch die Schüler.

Algorithmen & Programmieren für alle Fächer

Die Befähigung von Schülern zu vertieftem algorithmischen Denken bzw. konstruktivem Problemlösen ist natürlich auch eine Grundforderung in allen Unterrichtsfächern. Dass Algorithmen im Alltag des 21. Jahrhunderts wichtig sind, sehen auch mathematikferne Schüler und Lehrer spätestens dann ein, wenn man sie mit den uns allen bekannten Beispielen konfrontiert wie etwa dem Aufbauen eines Ikea-Möbels, dem Programmieren einer Videoaufnahme oder dem Backen eines Kuchens. Man vergleiche hierzu insbesondere die Ausführungen in Schmidt-Thieme 2005, wo über die große Bedeutung einer fächerübergreifenden Algorithmenausbildung vom Mathematik- bis zum Deutschunterricht berichtet wird.

Algorithmen & Programmieren: eine grundlegende IT-Kompetenz


Die Grundlage jeglicher moderner IT-Anwendungen sind seit jeher Algorithmen und Programmierung. Nur wer mit deren Basiskonzepten in Berührung gekommen ist, kann vertieft Chancen und Risiken unserer modernen Informationsgesellschaft beurteilen und die sich hierdurch neu ergebenden Möglichkeiten konstruktiv und kritisch einsetzen. Die Bandbreite der Beispiele ist nahezu unüberschaubar: Sie reicht vom Onlinebanking über gezielte Werbung, Expertensysteme in der Medizin, Suchmaschinen, Wirtschaftssimulationen, Wahlcomputern bis hin zu Computerspielen – eben alles was zwischen Einschalten und Ausschalten eines Computers passiert. Aus diesen Überlegungen resultiert naheliegenderweise die Forderung nach Basiskenntnissen in Algorithmen und Programmierung für jeden, der moderne IT-Systeme verwendet – und die ebenso naheliegende Frage, wie man dieses realisieren kann.

Warum gerade Scratch?

Überlegungen zum schulischen Programmieren sind nicht neu, beginnend mit Logo (1969) gibt es mit Karel the Robot (1981), Etoys und Squeak (1994), Lego Mindstorms (1998), Kara (2000), Greenfoot (2006) und vielen anderen eine große Anzahl von Programmierumgebungen. Diese Ansätze richten sich jedoch eher an den Anforderungen mathematischen bzw.

informatischen Unterrichts aus bzw. erfordern für Schüler eine längere Einarbeitungszeit oder für die Lehrkraft vertiefte Programmierkenntnisse.

Idealerweise sollte die Programmierumgebung für Schüler und Lehrer schon in der ersten Unterrichtsstunde produktive schulfachbezogene Ergebnisse ermöglichen. Für beide Zielgruppen steht hier im Vordergrund, dass es um Algorithmen und Programmierung geht – nicht um eine „Produktschulung“. Die Programmierumgebung an sich ist nur das Werkzeug zum Zweck. Jeglicher Aufwand zum Erlernen der Programmiersprache an sich wie auch zur Behebung von Syntaxfehlern sollte minimiert werden:

Java	Scratch
<pre>public class Hallo {public static void main (String argv[]) {System.out.println("Hallo Welt");}}</pre>	

Scratch bietet die Vorteile, dass Befehle selbsterklärend sind, Syntaxfehler nicht existieren, logische Fehler oft „(ein)gesehen“ werden und schnelle Erfolgsmöglichkeiten sowie hohe Schülermotivation Hand in Hand gehen.

Scratch wurde von der Lifelong Kindergarten research group am Massachusetts Institute of Technology 2007 veröffentlicht und steht für Windows, Mac OSX und Linux kostenlos zur Verfügung. Neben dem schulischen Unterricht (siehe etwa Romeike 2007) wird Scratch bzw. Modifikationen davon auch an Universitäten zur Einführung in die Programmierung verwendet (vergl. Resnick et. al. 2009 bzw. Harvey/Mönig 2010).

Algorithmen & Programmieren in der IuK-Ausbildung

Gemäß obigen Ausführungen lässt sich Scratch für den Bereich Algorithmen und Programmierung nicht nur im Mathematik- oder Informatikunterricht einsetzen, sondern könnte sich auch eignen, diesen Bereich als gewünschte IT-Basiskompetenz für alle Lehramtsstudierenden zu realisieren.

Das Pflicht-Computerpraktikum IuK ist an der Universität Hildesheim für alle Lehramtsstudierende im zweiten Studienjahr vorgesehen. Für die Realisierung der fächerübergreifenden und alltagsrelevanten Algorithmen- und Programmierausbildung wird Scratch an drei Terminen eingesetzt.

Beim ersten Termin erkennen die Studierenden nach einer theoretischen Einführung in grundlegende Konzepte an kleineren Beispielen selbst die Möglichkeiten von Scratch und setzen sie in ersten Programmierprojekten um. Beim zweiten Termin erstellen die Studierenden eigenständig ein eigenes Konzept zur Einführung in Scratch für Schulkinder – mit Bezug zu ihren jeweiligen Unterrichtsfächern. Ausgehend von ihren fachlichen und

fachdidaktischen Kenntnissen werden diverse Einstiege und Umsetzungen für konkrete Unterrichtssituationen konzipiert. Beim dritten Termin wird ihr Konzept gemeinsam in der Praxis mit einer Schulklasse erprobt, damit die Studierenden gemeinsam mit den Schülern den Startpunkt für nachhaltiges Lernen setzen können. Hierbei überrascht die Studierenden, dass es den Schülern oft leichter fällt als ihnen selbst, mit Scratch zu arbeiten. Die bisherigen Erfahrungen sind sowohl für Lehrende wie auch für Studierende sehr vielversprechend. Wir beabsichtigen daher in den kommenden Semestern diesen Ansatz vertieft zu erproben und werden hierüber an geeigneter Stelle ausführlich weiter berichten.

Literatur

- Bruder, R., Leuders, T., Büchter, A. (2008): Mathematikunterricht entwickeln. Bausteine für kompetenzorientiertes Unterrichten, Berlin: Cornelsen-Verlag.
- Feurzeig, W., Papert, S. (1969): Programming-languages as a conceptual framework for teaching mathematics. In: Programmed Learning Research: Paris: Dunod.
- Harvey, B., Mönig, J. (2010): Bringing "No Ceiling" to Scratch: Can One Language Serve Kids and Computer Scientists? Constructionism 2010 conference, Paris.
- Helmke, A. (2011): Mathematische Begabung - typisch Mädchen oder typisch Junge? In: Beiträge zum Mathematikunterricht 2011 (in Vorbereitung).
- Howe, J. A. M. et. al. (1988): Teaching Mathematics Through Programming in the Classroom. In: Soloway, E., Spohrer J.C. (Hrsg.): Studying the Novice Programmer. Hillsdale: L. Erlbaum Associates Inc.
- Kortenkamp, U. (2008): Strukturieren mit Algorithmen. In: Kortenkamp et. al. (Hrsg.): Informatische Ideen im Mathematikunterricht. Hildesheim: Franzbecker, 77-85.
- Modrow, E. (2006): Zur Ordnungswirkung fundamentaler Ideen der Informatik am Beispiel der theoretischen Schulinformatik. *informatica didactica* 6, 2006.
- Resnick, M. et. al. (2009): Scratch: Programming for All. *Communications of the ACM*, November 2009, 60-67.
- Romeike, R. (2007): Animationen und Spiele gestalten – ein kreativer Einstieg in die Programmierung. <http://www.funlearning.de/> (Zuletzt abgerufen am 11.3.11).
- Schmidt-Thieme, B. (2005): Algorithmen – fächerübergreifend und alltagsrelevant? In: Engel, Joachim u. a. (Hrsg.): Strukturieren - Modellieren - Kommunizieren. Leitbilder mathematischer und informatorischer Aktivitäten. Hildesheim 2005, 177-188.
- A. Schwill (1995): Fundamentale Ideen in Mathematik und Informatik. In: Bericht über die 12. Arbeitstagung des AK "Mathematikunterricht und Informatik" der GDM (H. Hischer, M. Weiß, eds.), 18-25.
- Strecker, K. M. (2009): Informatik für Alle – Wie viel Programmierung braucht der Mensch? Dissertation, Georg-August-Universität Göttingen.
- Tietze, U.-P., Klika, M., Wolpers, H. (1997): Mathematikunterricht in der Sekundarstufe II: Band 1. Braunschweig/Wiesbaden: Vieweg Verlagsgesellschaft.
- Weigand, H.-G. (1989): Algorithmen und Computer im Mathematikunterricht. In: Beiträge zum Mathematikunterricht 1989, 386 – 389.