

Ein Bild sagt mehr als tausend Worte – interaktive Visualisierungen in web-basierten Programmieraufgaben

Natalie Kiesler¹

Abstract: Aus unserem täglichen Leben ist die Informatik kaum noch wegzudenken. Das Smartphone, SmartHome und ein oder andere Wearable umgibt und begleitet uns im Alltag. Nur durch Programme und Computer-Codes sind diese Entwicklungen möglich geworden. Programmieren lernen ist allerdings kein Kinderspiel; und Programmieren lehren auch nicht. Daher sollen die Chancen und Potentiale online-basierter Lehr-/Lernangebote mit Blick auf interaktive Programm-Visualisierungen neu betrachtet werden. Dabei ist es das Ziel, Lernenden den Zugang zum Programmieren und das Verständnis von Quellcode zu erleichtern und bekannte Herausforderungen in selbstgesteuerten Lernprozessen zu adressieren.

Keywords: Programmierausbildung, interaktive Visualisierung, Hochschuldidaktik, E-Learning

1 Ziele und Bedeutung

Ziel des laufenden Forschungsprojekts ist die Unterstützung und Aktivierung von Informatik-Studierenden beim Erlernen grundlegender Programmierkonzepte begleitend zum curricularen Kursangebot. Die Förderung selbstgesteuerter Lernprozesse zum Aufbau von Programmierkenntnissen und –kompetenzen steht dabei im Fokus. Im Rahmen des entstehenden Online-Angebots soll neben der Rezeption von Inhalten, Interaktion und individuelles Feedback durch Übungsaufgaben möglich gemacht werden. Demnach können Lernende Animationen programmieren, während differenziertes Feedback erfolgt und Programmiersprachen-unabhängiger Pseudo-Code sichtbar wird. Das allgemein bekannte Sprichwort „Ein Bild sagt mehr als tausend Worte“ bringt einen entscheidenden Vorteil des Angebots auf den Punkt. Ein Bild kann maßgeblich zum Verständnis komplizierter Sachverhalte beitragen. Insofern scheint es naheliegend, die Visualisierung abstrakter Programmierkonzepte zusammen mit weiteren Maßnahmen zu nutzen, um das Verständnis zu fördern [BM11], [ZSS13].

2 Konzept

Im Rahmen des Forschungsprojekts sollen modulare Online-Lerneinheiten für Informatik-Studierende entstehen, die sowohl multimediale Lerninhalte als auch interaktive Übungsaufgaben zur Programmierung anbieten. Die Web-Anwendung ist als

¹ Hochschule Fulda, Fachbereich Angewandte Informatik, Leipziger Straße 123, 36037 Fulda,
natalie.kiesler@informatik.hs-fulda.de

Blended Learning-Szenario oder Online-Szenario denkbar, in dem Studierende auf freiwilliger Basis teilnehmen. Die Lerneinheiten werden entlang didaktischer Variablen konzipiert, um sich a Lernende und curriculare Rahmenbedingungen anzupassen [Wi02]. Jede Lerneinheit enthält als essentiellen Bestandteil interaktive Übungsaufgaben, zu deren Lösung ausführbare, visuelle Code-Blöcke angeboten werden. Diese sind durch HTML, CSS, JavaScript und Google Blockly [Go16] implementiert.

Anders als bei bestehenden Angeboten wie etwa Scratch [Sc16] wird während der Zusammenstellung einzelner Blöcke nach dem Baukastenprinzip zeitgleich Pseudo-Code generiert, um die Struktur des Codes sichtbar zu machen. Für jeden Baustein ist modularer Pseudo-Code vordefiniert. Bei der Ausführung erfolgt eine je nach Blockzusammensetzung unterschiedliche dynamische Animation in Kombination mit textueller Rückmeldung. Aktuell ist jede Visualisierung und Rückmeldung einzeln mittels JavaScript ausprogrammiert. Ziel ist es, Fehler leichter nachvollziehbar zu machen und Lernende schrittweise zum Erkennen der Lösung anzuleiten. Durch die generierte Rückmeldung soll das Problem der Frustration aufgrund fehlenden Feedbacks gemindert werden. Feedback als extrinsischer Motivator kann eine wichtige Einflussgröße auf die Zufriedenheit der Lernenden darstellen [Ke00]. Zwar hat Computer-generiertes Feedback wohl weniger Einfluss als menschliches Feedback [HT07], doch ist die Bedeutung von Rückmeldeprozessen im Zusammenhang mit interaktiven Programmieraufgaben noch nicht hinreichend untersucht [HB07], [Ko09]. Auch die Frage, durch welche Maßnahmen intelligentes, visuelles Feedback in die Lehre von Programmierkompetenzen integriert werden kann, bleibt zu beantworten.

3 Aktueller Prototyp

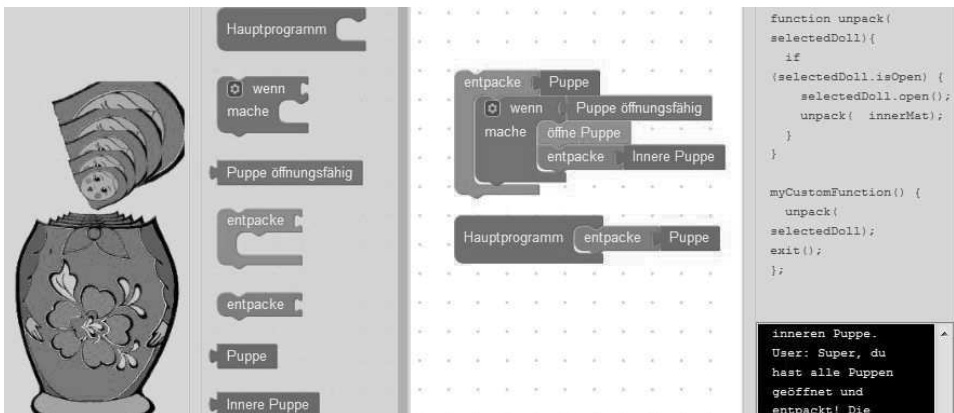


Abb. 1: Ausschnitt einer Bildschirmaufnahme der aktuellen Beispielaufgabe

Anhand der Abbildung ist es möglich, den Aufbau einer prototypischen Übung zur Rekursion nachzuvollziehen. Dazu ist die Seite in drei Bereiche aufgeteilt: Links ist die

Visualisierung zu finden, in der Mitte der Arbeitsbereich mit Code-Blöcken und auf der rechten Seite der generierte Pseudo-Code mit einem Bereich für System-generierte Rückmeldung. Im Beispiel sollen die Code-Blöcke so arrangiert werden, dass sowohl eine Funktion als auch dessen Aufruf in einem Hauptprogramm entstehen. Alle angezeigten Blöcke sind zur Lösung der Aufgabe notwendig, wobei variierende Schwierigkeitsgrade der Aufgabe durch eine größere Auswahl an Code-Blöcken realisiert werden können. Die Bausteine können ähnlich wie Puzzlestücke durch Drag-and-Drop intuitiv zusammengesetzt, verschoben und ggf. verworfen werden. Ein anfängliches Video-Tutorial führt die Lernenden in die Benutzung ein. Um die gegebene Aufgabe in Abbildung 1 zu lösen, müssen also alle notwendigen Bausteine zur Realisierung der rekursiven Lösung zusammengesetzt werden: Eine Funktion muss geschrieben, eine Abbruchbedingung festgelegt und ein Aufruf der Funktion durch sich selbst mit adäquatem Parameter festgelegt werden. Auch bei falscher Block-Zusammensetzung wird der Benutzer-generierte Code ausgeführt. Aktuell sind ca. 100 Visualisierungen und Rückmeldungen für fehlerhafte Block-Kombinationen als Ergebnis von Nutzertests implementiert. Die Erweiterung um weitere Komplexitätsgrade und eigenständige Programmieraufgaben via Quellcode-Eingabe ist geplant.

Literaturverzeichnis

- [BM11] Bentråd, S.; Meslati, D.: Visual Programming and Program Visualization - Towards an Ideal Visual Software Engineering System. In (ACEEE): Proceedings of International Conference in Communication and Information Technology 2011, S. 43-49, 2011.
- [Go16] Google Developers: Blockly, URL: <https://developers.google.com/blockly/>, Abrufdatum: 17.04.2016.
- [HT07] Hattie, J.; Timperley, H.: The Power of Feedback. In: Review of Educational Research 77.1/2007, S. 81–112, 2007.
- [HB07] Hundhausen, C.; Brown, J. L.: An experimental study of the impact of visual semantic feedback on novice programming. In: Journal of Visual Languages and Computing, 18/2007, S. 537–559, 2007.
- [Ke00] Keller, J.: How to integrate learner motivation planning into lesson planning: The ARCS model approach. VII Semanario, Santiago, Cuba, 2000.
- [Ko09] Kohl, L.: Kompetenzorientierter Informatikunterricht in der Sekundarstufe I unter Verwendung der visuellen Programmiersprache Puck. Jena, 2009.
- [Sc16] Scratch. URL: <https://scratch.mit.edu/> Abrufdatum: 27.01.2016.
- [Wi02] Wildt, J.: Ein hochschuldidaktischer Blick auf Lehren und Lernen. In (Berendt, B.; Voss H.-P.): Neues Handbuch Hochschullehre A 1.1., Raabe, Berlin, 2001.
- [ZSS13] Zhang, Y.; Surisetty, S.; Scaffidi, C.: Assisting comprehension of animation programs through interactive code visualization. In: Journal of Visual Languages and Computing, 24/ 2013, S. 313–326, 2013.