

Anwendungsorientierte Gestaltung eines Informatik-Vorkurses als Studienmotivator

Jan Holz, Nadine Bergner, Ulrik Schroeder

Lehr- und Forschungsgebiet Informatik 9: Learning Technologies

RWTH Aachen

Ahornstraße 55

52074 Aachen

holz@informatik.rwth-aachen.de

bergner@informatik.rwth-aachen.de

schroeder@informatik.rwth-aachen.de

Abstract: Zur Unterstützung von Studierenden in der Studieneingangsphase wurde an der RWTH Aachen ein neuartiger und motivierender Einstieg in den Vorkurs Informatik entwickelt und zum Wintersemester 2011/12 erprobt. Dabei wurde die grafische Programmierung mittels App Inventor eingeführt, die zur Umsetzung anwendungsbezogener Projekte genutzt wurde. In diesem Beitrag werden die Motivation für die Neugestaltung, das Konzept und die Evaluation des Testlaufs beschrieben. Diese dienen als Grundlage für eine vollständige Neukonzeption des Vorkurses für das Wintersemester 2012/2013.

1 Einleitung

Im Wintersemester 2011/2012 wurde an der RWTH Aachen im Rahmen des interdisziplinären Projekts „IGaDtools4MINT“¹ der Informatik-Vorkurs für Studierende durch das Lehr- und Forschungsgebiet Informatik 9 in Kooperation mit dem Lehr- und Forschungsgebiet Informatik 3 teilweise neu gestaltet. Darauf aufbauend erfolgt aktuell eine umfassende Neukonzeption des Vorkurses für das Wintersemester 2012/2013.

Der Vorkurs Informatik bietet Platz für 250 Teilnehmer/innen, wobei die Nachfrage noch darüber hinausgeht. Die Gruppe setzte sich 2011 heterogen aus Studierenden der Informatik (circa 50 %), Wirtschaftsingenieurwesen, Elektrotechnik, Mathematik, Biologie und vereinzelt weiteren Studiengängen zusammen. In seiner bisherigen Form umfasste der Vorkurs fünf Tage und findet Ende September statt. Am ersten Vormittag wurden zunächst organisatorische und infrastrukturelle Grundlagen geklärt. Anschließend erfolgte eine Einführung in Betriebssystem- und Shell-Bedienung. In den verbleibenden vier Tagen wurden in einem Wechsel aus Vorlesungs- und Übungsstunden die Grundlagen der Java-Programmierung vermittelt.

¹ Das Projekt IGaDtools4MINT wird vom Bundesministerium für Bildung und Forschung und dem Europäischen Sozialfonds für Deutschland gefördert. Informationen unter <http://www.igadtools4mint.de/> (08/2012).

Aufbauend auf dem bestehenden Vorkurs wurden, anschließend an die organisatorischen Grundlagen, zwei halbe Tage testweise neu gestaltet. Danach wurde zum regulären Java-Teil des ursprünglichen Vorkurses gewechselt.

2 Hintergrund

Vor dem Hintergrund hoher Studienabbruchzahlen in der Informatik, die vor allem auf Probleme in den ersten Semestern zurückzuführen sind, gilt es insbesondere die Studieneingangsphase zu betrachten und Studierende in dieser Zeit gezielt zu unterstützen [We10, He10, Kn11]. Zu diesem Zweck wird im Rahmen des Projekts IGaDtools4MINT ein mehrstufiges Förderkonzept zur Unterstützung des Übergangs von Schule zu Hochschule und der Studieneingangsphase in der Informatik entwickelt und umgesetzt [Ap12]. Im Folgenden wird das Vorkurs-Konzept, welches einen Brückenschlag von der Schule zur Universität darstellt [Sc03], detailliert beschrieben.

2.1 Ziele

Der Vorkurs verfolgt das Ziel, Erfahrungslücken bei den zukünftigen Studierenden zu schließen und sie auf den anstehenden Studienbeginn fachlich und methodisch vorzubereiten [BH10, PMO09]. Dazu sollen neben konkreten informatischen Inhalten vor allem grundlegende fachliche Konzepte und Arbeitsweisen erarbeitet werden [Sc93]. Um im Rahmen der Veranstaltung möglichst alle Studierenden zu erreichen, wurden bei der Konzeption speziell Gender- und Diversity-Aspekte berücksichtigt. Dies umfasst unter anderem das didaktische Konzept, die Wahl der Beispiele und die Formulierungen für die Arbeitsmaterialien. Dabei ist es das Ziel, die Studienmotivation der Teilnehmerinnen und Teilnehmer zu fördern, indem durchgängig Anwendungs- und Alltagsbezüge hergestellt werden. So wird der motivierende Schritt von der Nutzung zur Gestaltung von Informatikinhalten vollzogen, der das Interesse am Fach mitbestimmt [SK10]. Um das aktive und selbstgesteuerte Lernen zu unterstützen, wird in einem Großteil der Veranstaltung Peer-Teaching in Kleingruppen eingesetzt.

2.2 Wahl der Werkzeuge

Da ein Fokus des Vorkurses auf dem Verständnis von informatischen Konzepten und Arbeitsweisen liegt, sollen fachinhaltliche Barrieren, wie beispielsweise Syntaxprobleme bei der Programmierung, möglichst vermieden werden. Als Werkzeug für den Einstieg wird daher eine grafische Programmiersprache einer textuellen vorgezogen. Hierzu sind bereits Lernumgebungen wie Scratch² und Alice³ bekannt und erprobt [Ma10, MLC04]. Diese haben als abgeschlossene Mikrowelten jedoch den Nachteil, dass anstelle eines echten Anwendungs- und Alltagsbezugs eine simulierte und nicht-altersgerechte Spielwelt im Zentrum steht. Es bietet sich daher an, stattdessen die aktuelle Entwicklungs-

² <http://www.scratch.mit.edu/> (08/2012)

³ <http://www.alice.org/> (08/2012)

umgebung App Inventor⁴ einzusetzen, die es Programmieranfängern ermöglicht, in kurzer Zeit reale Produkte in Form einsetzbarer Applikationen für Android Smartphones zu gestalten [Mo11, Wo11]. Dadurch werden schnell erste praxisnahe Gestaltungserfolge in einem sozialen Kontext erreicht [MLC04]. Der Einsatz aktueller Technologien wirkt dabei zusätzlich motivierend für die zukünftigen Studierenden [De02]. Zu dem Einsatz der Smartphone-Programmierung in einer gender-gerechten Lernumgebung gibt es bereits Vorarbeiten für die gymnasiale Oberstufe, die hier fortgesetzt werden [HLS11].

2.3 Kurzvorstellung des App Inventors

Der App Inventor wurde seit Dezember 2010 von Google öffentlich als grafische Entwicklungsumgebung für Android Apps zur Verfügung gestellt. Ein Jahr später, im Dezember 2011, wurde das Angebot bei Google eingestellt, vom neu gegründeten MIT Center for Mobile Learning übernommen und ist dort seit März 2012 wieder öffentlich verfügbar.

Der App Inventor teilt sich in zwei Editoren: Den „Design Editor“ (Abbildung 1) für die Gestaltung der GUI sowie den „Blocks Editor“ (Abbildung 2), in dem die eigentlich Programmierung über grafische Bausteine erfolgt.

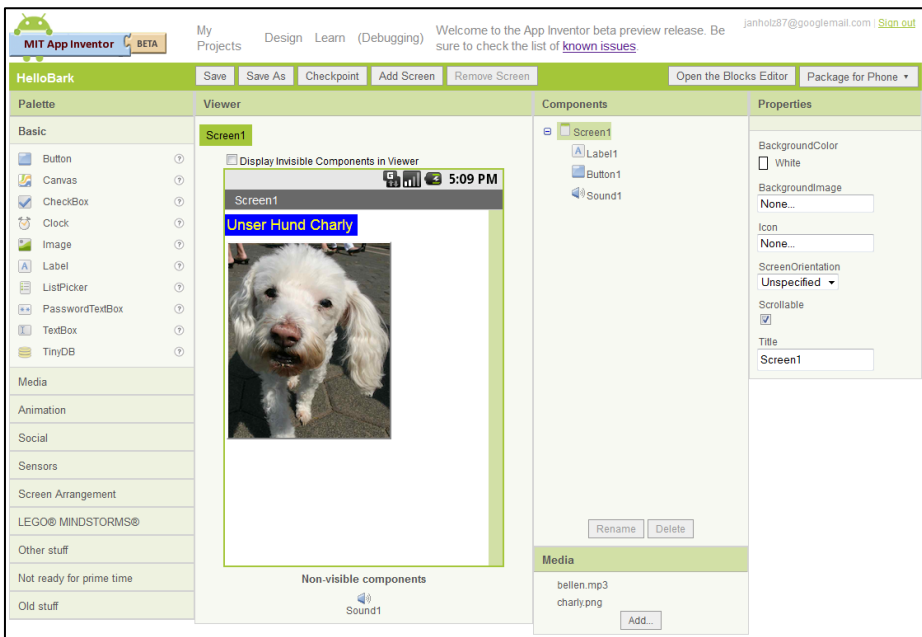


Abbildung 1: Screenshot des „Design Editor“ mit dem ersten eigenen Programm „Hello Bark!“

⁴ <http://www.appinventor.mit.edu/> (08/2012)

In beiden Editoren erfolgt das Hinzufügen und Entfernen von Komponenten durch einfache Drag'n'Drop-Aktionen. Im Design Editor können Elemente frei kombiniert, angeordnet und konfiguriert werden. Bei der Programmierung im Blocks Editor werden syntaktische Fehler vermieden, indem sich nur sinnvoll kombinierbare Bausteine miteinander verbinden lassen. Dabei lassen sich die Kombinationsmöglichkeiten durch Form und Farbe der Bausteine erkennen.

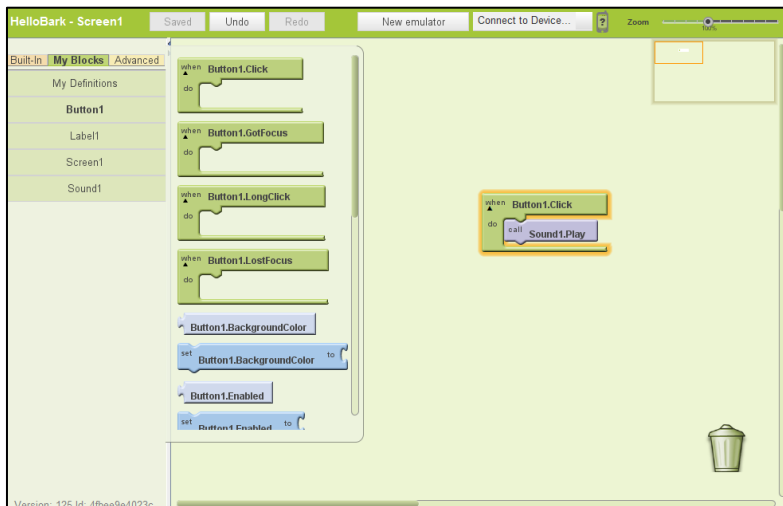


Abbildung 2: Screenshot des „Blocks Editor“ mit dem ersten eigenen Programm „Hello Bark!“

3 Prototypische Neugestaltung

Die testweise Neugestaltung des Vorkurses erstreckte sich über zwei Tage. Dabei entstand folgender Ablaufplan:

Tag 1: Zeiten	Inhalt	Sozialform
09:00 – 10:15	Begrüßung und Organisatorisches	Präsentation
10:15 – 10:45	Vorstellung des App Inventors	Präsentation
10:45 – 11:45	Infrastruktur einrichten und Arbeitsblatt „Hello Bark“ (Einführung in den App Inventor)	Kleingruppen, Peer Teaching
11:45 – 12:45	Mittagspause	
12:45 – 14:15	Arbeitsblätter „Hello Bark!“ und „Paint Pot“ (Variablen und Fallunterscheidungen)	Kleingruppen, Peer Teaching
14:15 – 14:45	Pause	
14:45 – 17:00	Arbeitsblatt „Paint Pot“ und Ergebnissicherung	Kleingruppen, Peer Teaching

Tabelle 1: Ablauf des ersten Tages im Vorkurs Informatik

Tag 2: Zeiten	Inhalt	Sozialform
09:00 – 09:30	Vorstellung der Ergebnisse von Tag 1	Präsentation
09:30 – 11:00	Arbeitsblätter „Collect Androids I“ und „Collect Androids II“ (Prozeduren und Schleifen)	Kleingruppen, Peer Teaching
11:00 – 11:15	Pause	
11:15 – 12:00	Arbeitsblatt „Collect Androids II“, Ergebnissicherung und Evaluation	Kleingruppen, Peer Teaching
12:00 – 13:00	Mittagspause	
13:00 – ...	Übergang zum gekürzten Programm des alten Informatik-Vorkurses	Vorlesungen und Übungen

Tabelle 2: Ablauf des zweiten Tages im Vorkurs Informatik

Wie dem Ablaufplan zu entnehmen ist, enthält der Vorkurs nur noch kurze frontale Phasen mit einführenden oder zusammenfassenden Präsentationen im Hörsaal. Schwerpunktmäßig finden ausführliche Gruppenarbeitsphasen in den Seminarräumen statt. Für diese wurde die Gruppe der 250 Teilnehmer/innen zunächst in 9 Kleingruppen mit jeweils einem betreuenden Tutor oder einer betreuenden Tutorin pro 20 Teilnehmenden unterteilt. Innerhalb dieser Gruppe wurden Zweier- bis Dreier-Teams gebildet, die gemeinsam die Aufgabenstellungen bearbeiteten. Die Tutoren nahmen dabei keine Lehrer- oder Dozentenrolle ein, sondern leiteten die Gruppen zum selbstständigen Arbeiten im Team und Peer Teaching an. Zudem standen die Tutoren natürlich für alle Fragen und Probleme zur Verfügung, die die Teilnehmer/innen nicht selbst klären oder lösen konnten.

3.1 Materialien

Im Folgenden werden die erstellten und verwendeten Materialien zur graphischen Programmierung vorgestellt:

- Informationsblatt: „*App Inventor – Grundlagen*“
- Arbeitsblatt „*Hello Bark!*“:
Hello World Programm unter Verwendung von Bild und Ton
- Arbeitsblatt „*Paint Pot*“:
Paint-Programm, um mit den Fingern auf dem Bildschirm zu zeichnen
- Arbeitsblatt „*Collect Androids I*“:
Reaktionsspiel bei dem Objekte durch Antippen eingesammelt werden
- Arbeitsblatt „*Collect Androids II*“:
Erweiterung des Reaktionsspiel um zusätzliche Funktionen

Die Aufgaben basieren inhaltlich auf dem Buch „App Inventor“ von David Wolber [Wo10], welches sich zum guten Teil mit den offiziell online verfügbaren Tutorials zum App Inventor deckt. Auf Grundlage dieser Materialien wurden eigene Varianten der Apps entwickelt, die jeweils bestimmte informatische Konzepte in den Mittelpunkt stellen, und in Arbeitsblättern neu aufbereitet.

Mit der App „Hello Bark!“ wird als Einstieg ein einfaches „Hello World“-Programm umgesetzt, bei dem durch das Drücken eines Buttons, welcher das Bild eines Hundes zeigt, eine mp3-Datei mit Hundegebell abgespielt wird. Im zweiten Projekt wurde eine einfache Paint-App entwickelt, bei der in verschiedenen Farben und Strichstärken mit den Fingern auf dem Bildschirm gezeichnet werden kann. Als letztes Projekt wurde in zwei Arbeitsblättern ein Reaktionsspiel erstellt, bei dem verschiedene Grafiken zeitlich begrenzt auf dem Bildschirm erscheinen und durch Antippen für Plus- oder Minuspunkte sorgen. In der Erweiterung des Spiels können eine Punkteanzeige, eine Siegerehrung und ein Geschwindigkeitsregler ergänzt werden.

Insgesamt werden mit den Arbeitsblättern folgende Lerninhalte abgedeckt:

- Variablen
- Fallunterscheidungen
- Schleifen
- Prozeduren

Die Vermittlung dieser Inhalte erfolgt im Rahmen der konkreten Produktentwicklung, sodass die Notwendigkeit der Konzepte unmittelbar erkannt werden kann. Zudem werden zentrale Aspekte in Übersichtsboxen zusammengefasst (Abbildung 3).

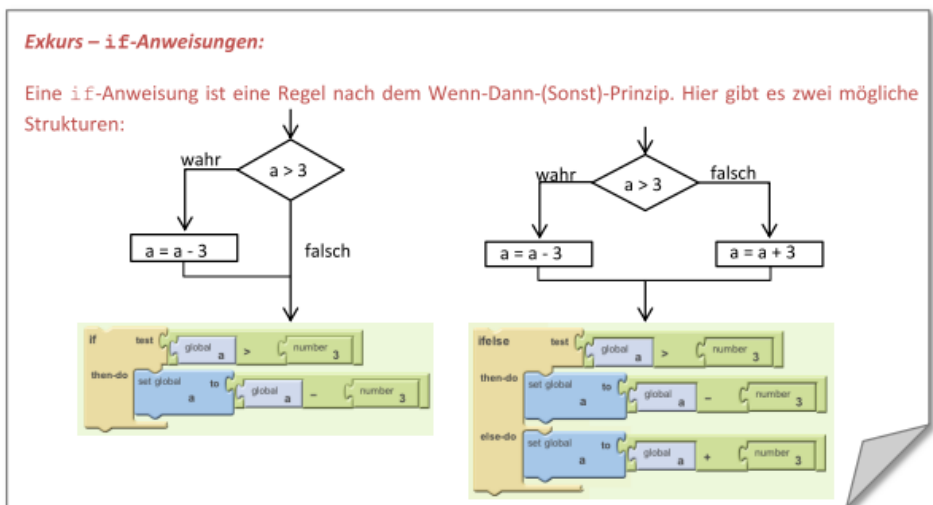


Abbildung 3: Ausschnitt aus dem Arbeitsblatt „Paint Pot“ – Beispiel für eine Übersichtsbox zu einem Programmierkonstrukt

Während die ersten Arbeitsblätter noch sehr konkrete Schritt-für-Schritt-Anweisungen zur Umsetzung der Projekte enthalten, sind die weiteren Arbeitsblätter offener gestaltet, sodass die Studierenden selbstständig auf bereits gelernte Inhalte zurückgreifen und die Freiheit haben, eigene Vorstellungen umsetzen. Insgesamt wurden die Arbeitsblätter zum App Inventor nach einem einheitlichen Schema aufgebaut:

- *Motivation*
Jedes Arbeitsblatt beginnt mit einer kurzen, motivierenden Übersicht über den Sinn und Inhalt der Aufgabe:
 - Was wird mit diesem Arbeitsblatt erarbeitet?
 - Welches Produkt mit welchen Funktionen wird am Ende erreicht?
 - Was sind die Lernziele?
- „Das Layout – das Aussehen der App“
Schrittweise werden die benötigten GUI-Komponenten eingeführt und im Editor zusammengesetzt.
- „Der Blocks Editor – die Funktionen der App“
Unterteilt in mehrere Schritte werden die einzelnen Komponenten der GUI mit den entsprechenden Funktionen versehen. Dabei werden die jeweils benötigten Programmierkonstrukte (Fallunterscheidungen, Schleifen, ...) eingeführt. Die verschiedenen Komponenten werden durch unterschiedliche Textformatierungen hervorgehoben (Abbildung 4).

Für diese Inhalte wurde mit Hilfe des App Inventors ein erstes grundlegendes Verständnis geschaffen, das im anschließenden Java-Teil des Vorkurses weiter vertieft und gefestigt wurde. Dazu wurden die bestehenden Materialien des alten Vorkurses um Querverweise auf die Lerninhalte des App-Inventor-Teils ergänzt.

Schritt 2 – Aufruf der Prozedur BewegeAndroid

Nun soll die selbst definierte Prozedur auch zum Einsatz kommen. Da die App zu Beginn das Männchen zufällig positionieren soll, ist es sinnvoll, die vordefinierte Funktion `Screen1.Initialize` zu benutzen.

- 1) Wir ziehen den Block `Screen1.Initialize` unter My Blocks in Screen1 auf die Fläche. Beim Initialisieren soll eine zufällige Position bestimmt werden. Also wählen wir unter My Blocks in My Definition unsere eigens kreierte Funktion `call BewegeAndroid` aus und setzen sie in die Funktion.
- 2) Damit sich die Position unseres Männchens jede Sekunde ändert, benutzen wir Clock Komponente. Dazu erstellen wir nun das bereits oben erwähnte `Clock1.Timer` Event, in der wir unsere Funktion `BewegeAndroid` aufrufen.






Abbildung 4: Ausschnitt aus dem Arbeitsblatt „Collect Androids I“ – Beispiel für einen Arbeitsschritt im Blocks Editor

3.2 Einschränkungen

Durch die Größe des Vorkurses ergibt sich zunächst der Nachteil, dass die Teilnehmenden nicht, wie in den kleineren Workshops unseres Informatik Schülerlabors Info-

Sphere⁵, von uns mit Smartphones ausgestattet werden können, sondern auf einen Emulator zurückgegriffen werden muss. Durch die zunehmende Verbreitung geeigneter Smartphones in der Zielgruppe können viele Teilnehmer jedoch auf ein eigenes Gerät zurückgreifen [ji11]. Da zu erwarten ist, dass sich dieser Trend für Smartphones und auch Tablets fortsetzen wird, besteht die Möglichkeit, den Emulator in Zukunft als Ersatzlösung zu betrachten, da zudem Geräte für bis zu 40 Arbeitsgruppen zur Verfügung gestellt werden können. Dadurch wird sich das Spektrum möglicher Aufgaben erweitern, da Einschränkungen des Emulators (beispielweise bei der Nutzung von Sensoren) wegfallen.

4 Evaluation

Der neugestaltete Teil des Vorkurses wurde vor dem Übergang zum regulären Vorkurs durch einen Online-Fragebogen evaluiert, bei dem den Studierenden Gelegenheit für Feedback geboten wurde. Die Erstellung des Fragebogens erfolgte in Kooperation mit dem Lehr- und Forschungsgebiet „Gender und Diversity in den Ingenieurwissenschaften“ der RWTH Aachen. Der Fragebogen wurde von 152 Studierenden vollständig ausgefüllt (198 aktive Teilnehmer, Rücklaufquote 77 %). Innerhalb des Vorkurses lag die Frauenquote mit 29 %. Im Vergleich dazu liegt die Frauenquote im Studiengang Informatik dieses Jahrgangs bei 11 % [rw11].

Die Einschätzung des Vorkurses durch die Teilnehmer/innen ist in Tabelle 3 zusammengefasst.

	männlich	weiblich
Kursniveau angemessen	75%	50%
Inhaltliche Verbindung zu anderen Fächern erkennbar	3%	9%
Anwendungsbezug für die Praxis ersichtlich	51%	36%
Unterrichtssprache auch ohne Fachvokabular verständlich	84%	75%
Auf Rückfragen wurde eingegangen	91%	80%
Einführung App Inventor als gute Vorbereitung empfunden	71%	61%
Frontale Lerneinheiten zu gering	14%	14%

Tabelle 3: Einschätzung des Vorkurses

Die inhaltliche Verbindung zu anderen Fächern wurde als sehr schwach angesehen und wird in der aktuellen Konzeption des nächsten Vorkurses stark ausgebaut. Ebenso soll durch stärker miteinander verknüpfte Projektaufgaben der Anwendungsbezug innerhalb des Vorkurses noch weiter gestärkt werden. Das Lehr-Lern-Konzept mit der zuvor beschriebenen Aufteilung in frontale Phasen und Gruppenarbeitsphasen wird nach dem positiven Ergebnis der Evaluation beibehalten. Insgesamt wurde der neue Teil des Vorkurses von 72 % der Teilnehmer/innen mit „sehr gut“ oder „gut“ bewertet (Abb. 5):

⁵ <http://www.schuelerlabor-informatik.de/> (08/2012)

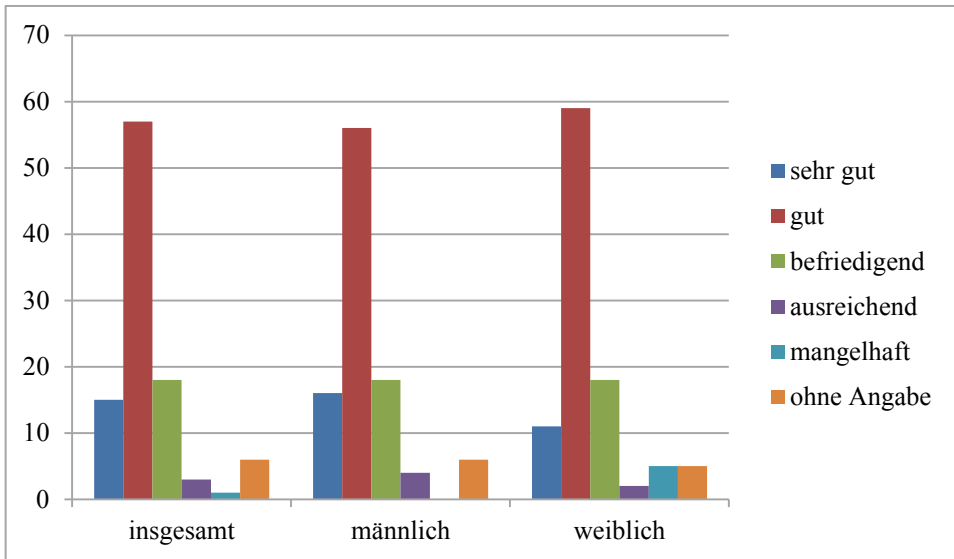


Abbildung 5: Benotung des neugestalteten Vorkurs-Teils (Angaben in Prozent)

Insbesondere von Interesse waren Freitext-Antworten zur zusammenfassenden Einschätzung des neuen Vorkurs-Teils. Hierbei wurde deutlich, dass der Zugang über den App Inventor vor allem von Studierenden mit wenigen Vorkenntnissen positiv aufgenommen wurde:

„Angenehmer Lernfortschritt, bei dem selbst ohne jegliche Vorkenntnisse gutes Mitarbeiten möglich ist. Bisher behandelte Themen und Arbeitsmethoden wecken weiteres Interesse. Sehr angenehmes Arbeitsklima“

„Die Aufgaben machen Spaß und sind zwar teilweise schwer aber trotzdem zu meistern. Durch die Erfolge im Vorkurs fühle ich mich für das Studium ermutigt.“

„Erste Berührungspunkte mit der Programmierung wurden abgebaut.“

„Der Vorkurs bietet einen seichten Einstieg, ohne dabei zu verheimlichen, dass kommende Aufgaben dennoch anspruchsvoll werden. Man hat jedoch schon jetzt die Gelegenheit, eine erfolgreiche Lern- und Übungsstrategie zu entwickeln.“

Studierende mit mehr Vorkenntnissen hatten zwar Spaß an der App-Entwicklung, wollten aber schneller zur textuellen Programmierung wechseln:

„Weniger App Inventor, schneller zur "Sprache".“

„Auf App Inventor verzichten und direkt eine "richtige" Programmiersprache wie Java, C, Ada, Basic etc. einführen.“

Somit wurde jedoch die eigentliche Zielgruppe, Studierende mit fehlenden Vorkenntnissen, durch den neu gestalteten Teil des Vorkurses erreicht.

Nach der vollständigen Umgestaltung wird im Wintersemester 2012/2013 eine ausführliche Evaluation des komplett überarbeiteten Vorkurses erfolgen.

5 Ausblick

Auf Basis der gewonnenen Erkenntnisse wird eine vollständige Neukonzeption des Informatik-Vorkurses für das Wintersemester 2012/2013 vorgenommen. Der Vorkurs wird dabei von fünf auf acht Tage erweitert, um selbstständiges, selbstbestimmtes und aktives Erarbeiten der Inhalte in Kleingruppen noch stärker zu ermöglichen.

Es sollen weiterhin der App Inventor und Java eingesetzt werden, wobei die Einführung zunächst über den App Inventor erfolgt. Bei der Gestaltung der Arbeitsblätter werden nach den bisherigen Erfahrungen die jeweiligen Lernziele klarer herausgestellt. Zudem wird zukünftig deutlicher durch gezielte Fragestellungen auf eine abschließende Ergebnissicherung hingearbeitet. Des Weiteren werden für die Neukonzeption die Inhalte von grafischer und textueller Programmierung stärker miteinander verbunden, indem beispielsweise Bausteine des App Inventors direkt zu Java Code in Bezug gesetzt werden.

Inhaltlich wird sich der neue Kurs vor allem auf Konzepte, Denk- und Arbeitsweisen der Informatik stützen, die anhand von interdisziplinär anwendungsbezogenen und gender- und diversity-gerechten Programmierprojekten zur Vorbereitung auf das erste Semester vermittelt werden.

Literatur

- [Ap12] Apel, R., Berg, T., Bergner, N., Chatti, M. A., Holz, J., Leicht-Scholten, C., Schroeder, U.: Ein vierstufiges Förderkonzept für die Studieneingangsphase in der Informatik. In (Forbrig, P., Rick, D., Schmolitzky, A. Hrsg.): *HDI 2012 – Informatik für eine nachhaltige Zukunft*. Universitätsverlag Potsdam, 2013; S. 143–148 (in diesem Band).
- [BH10] Berges, M., Hubwieser, P.: Vorkurse in objektorientierter Programmierung: Lösungsansatz für ein Problem der Einführungsveranstaltungen. In (Engbring, D., Keil, R., Magenheimer, J., Selke, H. Hrsg.): *HDI2010 – Tagungsband der 4. Fachtagung zur „Hochschuldidaktik Informatik“*. Universitätsverlag Potsdam, 2010; S. 13–22.
- [De02] Deinmann, M.: Motivationale Bedingungen beim Lernen mit Neuen Medien. In (Bleek, W.-G., Krause, D., Oberquelle, H., Pape, B.): *Medienunterstütztes Lernen – Beiträge von der WissPro-Wintertagung 2002*. Fachbereich Informatik, Universität Hamburg, 2002; S. 61–70. – URL [http://edoc.sub.uni-hamburg.de/informatik/volltexte/2009/53/\(12/2012\)](http://edoc.sub.uni-hamburg.de/informatik/volltexte/2009/53/(12/2012))
- [He10] Heublein, U., Hutzsch, C., Schreiber, J., Sommer, D., Besuch, G.: Ursachen des Studienabbruchs in Bachelor- und in herkömmlichen Studiengängen. Ergebnisse einer bundesweiten Befragung von Exmatrikulierten des Studienjahres 2007/08. HIS: Forum Hochschule, 2010. – URL http://www.his.de/pdf/pub_fh/fh-201002.pdf (12/2012)

- [HLS11] Holz, J., Leonhardt, T., Schroeder, U.: Using smartphones to motivate secondary school students for informatics. In: *Proceedings of 11th Koli Calling International Conference on Computing Education Research*. ACM, New York, 2011; S. 89–94. – DOI: 10.1145/2094131.2094148
- [ji11] JIM-Studie 2011: Jugend, Information, (Multi-)Media : Basisuntersuchung zum Medienumgang 12- bis 19-Jähriger. Medienpädagogischer Forschungsverbund Südwest, Stuttgart, 2011. – URL <http://www.mpfs.de/index.php?id=225> (08/2012)
- [Kn11] Knobelsdorf, M.: *Biographische Lern- und Bildungsprozesse im Handlungskontext der Computernutzung*. Dissertationsschrift. Fachbereich Mathematik und Informatik, Freie Universität Berlin, 2011. – URL http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_00000023809 (08/2012)
- [Ma10] Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E.: The Scratch Programming Language and Environment. In: *ACM Transactions on Computing Education* 10(4), 2010; art. 16. – DOI: 10.1145/1868358.1868363
- [MLC04] Moskal, B., Lurie, D., Cooper, S.: Evaluating the Effectiveness of a New Instructional Approach. In: *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '04)*. ACM, New York, USA, 2004; S. 75–79. – DOI: 10.1145/1028174.971328
- [Mo11] Morelli, R., de Lanerolle, T., Lake, P., Limardo, N., Tamotsu, B., Uche, C.: Can Android App Inventor Bring Computational Thinking to K-12? Unpublished manuscript. HFOSS 2010. – URL http://www.hfoss.org/uploads/docs/appinventor_manuscript.pdf (08/2012)
- [PB06] Pedroni, M., Bertrand, M.: The Inverted Curriculum in Practice. In: *Proceedings of the 37th SIGCSE technical symposium on Computer science education (SIGCSE '06)*, ACM, New York, 2006; S. 481–486. – DOI: 10.1145/1124706.1121493
- [PMO09] Pedroni, M., Meyer, B., Oriol, M.: What Do Beginning CS Majors Know? Report. Swiss Federal Institute of Technology, 2009. – DOI: 10.3929/ethz-a-006835722
- [rw11] RWTH Aachen University. *Zahlenspiegel 2011*. – URL http://www.rwth-aachen.de/global/show_document.asp?id=aaaaaaaaaacnrws (08/2012)
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik. In: *Zentralblatt für Didaktik der Mathematik* 1, 1993. – Auch online unter URL <http://ddi.cs.uni-potsdam.de/Forschung/Schriften/ZDM.pdf> (08/2012)
- [Sc03] Schinzel, B.: Curriculare Vorschläge zur Erhöhung des Frauenanteils in der Informatik – Möglichkeiten und Maßnahmen. Report. Albert-Ludwigs-Universität Freiburg, 2003. – URL <http://www.careerbench.uni-freiburg.de/cms/fileadmin/publikationen/curriculuminf.pdf> (12/2012)
- [SK10] Schulte, C., Knobelsdorf, M.: „Jungen können das eben besser“ – Wie Computernutzungserfahrungen Vorstellungen über Informatik prägen. In (Koreuber, M., Hrsg.): *Geschlechterforschung in Mathematik und Informatik. Eine (inter)disziplinäre Herausforderung*. Nomos Verlag, Baden-Baden, 2010; S. 87–110.
- [We10] Weihe, K.: Studieneingangsphase: Einsichten und Empfehlungen des Fakultätentags Informatik. Bericht. TU Darmstadt, 2010.
- [Wo10] Wolber, D., Abelson, H., Sperlus, E., Looney, L.: *App Inventor: Create Your Own Android Apps*. O'Reilly Media, Sebastopol, CA, USA, 2010.
- [Wo11] Wolber, D.: App inventor and real-world motivation. In: *Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11)*. ACM, New York, 2011; S. 601–606.