

Fünf Argumente für einen grafischen ProgrammierEinstieg - eine Studie über vier Jahrgangsstufen

Nadine Bergner¹, Thimo Leonhardt¹ und Ulrik Schroeder¹

Abstract: Grafische Programmierwerkzeuge sind ein beliebter Weg mit Kindern und Jugendlichen in die Programmierung einzusteigen. Ein grafischer Einstieg soll die Schülerinnen und Schüler altersgerecht ansprechen und ihnen aufgrund der Vermeidung von Syntaxfehlern erleichtern, eigene Projekte schnell und kreativ umzusetzen und dadurch motivierend wirken. In diesem Artikel werden die vermuteten Chancen und auch mögliche Gefahren eines grafischen ProgrammierEinstiegs wissenschaftlich untersucht. Dazu werden fünf Argumente beleuchtet: Begeisterungsfähigkeit, Vermittlung eines interessanten Bildes der Disziplin, Erwartungen der ProgrammierEinsteiger, Einfluss auf die wahrgenommene Schwierigkeit der Disziplin und Vermittlung eines kreativen Berufsbildes "Informatiker-in". Um diese Argumente wissenschaftlich zu untersuchen, wurden in 2015 und 2016 über 700 Kinder und Jugendliche vor und nach einem Einstiegsworkshop mit den Werkzeugen "Scratch" und "App Inventor" mittels Online-Fragebögen befragt. Die Ergebnisse decken auf, welche Auswirkungen die Workshops auf das Bild der Informatik sowie die Vorstellungen über Informatikerinnen und Informatiker haben.

Keywords: Grafische Programmierung, quantitative Evaluation, Scratch, App Inventor, Fragebogenstudie, Schülerlabor, Bild der Informatik, Berufsbild.

1 Motivation

Der Einstieg in die Programmierung kann inner- und außerhalb der Schule sowie in unterschiedlichen Altersstufen stattfinden. Den unterschiedlichen Gegebenheiten gemein ist, dass ein Weg gefunden werden muss, Kindern und Jugendlichen die Grundlagen der Programmierung zielgruppengerecht zu vermitteln.

Ein auch in Deutschland weit verbreiteter Ansatz ist die grafische Programmiersprache bzw. -umgebung *Scratch*. Scratch lässt den Lernenden Freiheiten mittels grafischer Programmierbausteine eigene interaktive Geschichten oder Spiele zu entwickeln, in denen Objekte bewegt werden können, Musik abgespielt und sogar mit dem Nutzer interagiert werden kann. Besonders durch das umfangreiche Angebot an ansprechenden und kindgerechten Grafiken werden Kinder der Klassenstufen 4 bis 8 mittels Scratch motiviert ihre eigenen Ideen in kleinen Programmen selbstständig zu verwirklichen. Das Werkzeug Scratch ist didaktisch sehr gut reduziert, kann durch das kindliche Design allerdings für ältere Jugendliche demotivierend wirken (vgl. [Ma10], [Ut10]).

¹RWTH Aachen, Informatik 9 - Lerntechnologien und Fachdidaktik Informatik, 52074 Aachen, {bergner, leonhardt, schroeder}@informatik.rwth-aachen.de

Eine weniger spielerische Möglichkeit des grafischen ProgrammierEinstiegs stellt der *App Inventor* des MIT bereit. Hier werden keine Katzen in bunten Unterwasserwelten, sondern Android-Apps für Smartphones und Tablets programmiert. Durch den Alltagsbezug der mobilen Geräte können Kinder und Jugendliche in einem sehr breiten Altersspektrum angesprochen und motiviert werden (vgl. [Wo11]).

Insbesondere durch den Bedeutungsgewinn der Programmierung für die Allgemeinbildung der Kinder könnte sich der Einsatz dieser Werkzeuge zukünftig verstärken. Offen bleibt dabei die Frage nach den Effekten solch grafischer ProgrammierEinstiege, welche Untersuchungsgegenstand dieses Artikels sind.

2 Forschungsfragen & -design

Beim Einsatz von Lernwerkzeugen stellt sich immer die Frage des Effekts sowie des Erfolgs bei der Zielgruppe, also den Lernenden. Dazu werden in diesem Paper die folgenden fünf Argumente, welche sich aus der oben genannten Literatur wie aus den Erfahrungen im Schülerlabor InfoSphere ableiten lassen, mittels einer Fragebogenstudie zu Einstiegsworkshops mit den Werkzeugen “Scratch” und “App Inventor” diskutiert:

1. Ein grafischer ProgrammierEinstieg kann die Lernenden für die Informatik begeistern.
2. Grafische Programmierung vermittelt das Bild einer interessanten und spannenden Wissenschaft Informatik.
3. Grafische Programmierung trifft die Erwartungen der ProgrammierEinsteiger.
4. Der grafische ProgrammierEinstieg hat Einfluss auf die wahrgenommene Schwierigkeit der Disziplin.
5. Grafische Programmierung vermittelt ein kreatives Berufsbild “Informatiker-in”.

Um Kinder und Jugendliche an die Informatik und speziell den Bereich der Programmierung heranzuführen, werden im Informatik-Schülerlabor InfoSphere an der RWTH Aachen seit 2010 über 30 verschiedene (in der Regel eintägige) Workshops angeboten, an denen Schulklassen und Kurse kostenfrei teilnehmen. Das Themenangebot beinhaltet Module für den Erstkontakt in der Grundschule und Unterstufe wie auch Module zur Vertiefung bzw. Anknüpfung an den bestehenden Informatikunterricht in der Mittel- und Oberstufe. Die Themen umfassen Facetten und Aspekte der theoretischen (z. B. zu Kryptographie-Verfahren), technischen (z. B. Schaltungen mit Mikrocontrollern) und praktischen (z. B. Erwerb von Programmierkompetenzen in unterschiedlichen Altersstufen) Informatik. Damit das punktuell angestoßene Interesse weiter ausgebaut werden kann, werden Kindern und Jugendlichen privat (an Wochenenden, in den Ferien) Möglichkeiten der tiefergehenden Einarbeitung geboten.

In dieser Auswertung bilden drei Module zum grafischen ProgrammierEinstieg für die Zielgruppe der Programmieranfänger der Klassenstufen 6 bis 9 den Fokus.

Das Forschungsdesign zur Evaluation der obigen fünf Argumente besteht aus einer quantitativen Forschungsmethodik mittels Online-Fragebögen², wobei der Pretest etwa ein bis zwei Wochen vor dem Besuch des Workshops in der Schule ausgefüllt wird und der Posttest direkt im Anschluss erhoben wird. Im Folgenden wird erläutert, mit welchen Fragen die fünf Aspekte beleuchtet werden.

Argument 1 - Begeisterungsfähigkeit

In dem Online-Fragebogen werden die Teilnehmerinnen und Teilnehmer explizit gefragt, wie ihnen der Einstieg in die (grafische) Programmierung gefallen hat und ob sie wieder an einem solchen Workshop teilnehmen möchten. Dazu steht den Lernenden eine fünfstufige Skala von „trifft gar nicht zu“ bis „trifft voll zu“ zur Beantwortung zur Verfügung. Analog wird ihre Zustimmung zur Aussage „Ich werde mich in Zukunft verstärkt mit Themen der Informatik beschäftigen.“ abgefragt.

Argument 2 - Vermittlung eines interessanten Bildes der Disziplin

Die Kinder und Jugendlichen werden vor und nach dem Besuch im InfoSphere dazu befragt, welches Bild der Disziplin Informatik sie haben. Dies geschieht zum einen, indem sie im Freitext drei Begriffe zur Informatik notieren. Zum anderen halten sie ihre Einstellung zu verschiedenen vorgegebenen Aussagen über die Informatik fest, indem sie einen Schieberegler zwischen zwei Gegenpositionen platzieren. Die hier relevanten Gegenpole sind „Informatik ist langweilig versus spannend.“, „Informatik ist interessant versus uninteressant.“ und „Informatik ist eintönig versus abwechslungsreich.“.

Argument 3 - Passung zu den Erwartungen der ProgrammierEinsteiger

Hier werden die Schülerinnen und Schüler auf einer fünfstufigen Skala danach befragt, wie exakt der Workshop ihren Erwartungen entsprochen hat. Darüber hinaus werden die Freitextangaben zu den Erwartungen im Pretest und die Verbesserungsvorschläge zum Workshop und die Themenwünsche für die Zukunft im Posttest ausgewertet.

Argument 4 - Einfluss auf die wahrgenommene Schwierigkeit der Disziplin

Für dieses Argument werden Angaben vor und nach dem Workshop verglichen. Relevant ist die Einschätzung zur Aussage „Informatik ist schwierig versus einfach.“.

Argument 5 - Vermittlung eines kreativen Berufsbildes “Informatiker-in”

Nach derselben Methodik wird die Vorstellung über Informatikerinnen bzw. Informatiker vor und nach dem Workshop erhoben. So können die Befragten den Schieberegler zur Aussage „Informatiker-innen müssen kreativ sein.“ nahezu stufenlos von „stimme gar nicht zu“ bis „stimme voll zu“ platzieren. Gleiches gilt für die Aussage „Informatiker-innen kreieren viel Neues.“.

² Weitere Details zum Forschungsdesign und dem konkreten Fragebogen können eingesehen werden unter [Be15].

3 Umsetzung der Workshops

Um eine wissenschaftliche Reflexion der Antworten der Teilnehmerinnen und Teilnehmer zu ermöglichen, werden in diesem Abschnitt die beiden verwendeten Werkzeuge „Scratch“ und „App Inventor“ wie auch die Ausgestaltung der Workshops im Schülerlabor InfoSphere beleuchtet.

3.1 Vergleich der Tools Scratch & App Inventor

Beide Werkzeuge zum grafischen ProgrammierEinstieg werden kostenlos, browserbasiert angeboten. Eine wichtige Gemeinsamkeit für diesen Forschungsansatz sind die optisch ähnlichen Programmierblöcke (siehe Abb. 1 für den Vergleich grundlegender Programmierbausteine), wobei speziell für jüngere Lernende die Option der breiten Sprachwahl (inklusive Deutsch) im Werkzeug „Scratch“ einen Vorteil gegenüber dem „App Inventor“ darstellt, welcher nicht in Deutsch verfügbar ist.



Abb. 1: Vergleich der grafischen Programmierbausteine (links Scratch, rechts App Inventor)

Im Gegensatz zu Scratch, welches 2007 speziell für Kinder entwickelt wurde, sollte der App Inventor ursprünglich Informatikkurse an Universitäten unterstützen und startete dort 2009 seine Pilotphase. Ein weiterer Unterschied ist, dass bei Scratch Objekte in einer Mikrowelt programmiert werden, welche direkt neben der Programmierfläche sichtbar ist, so dass die Lernenden stets in der gleichen Ansicht arbeiten. Im App Inventor gibt es dagegen zwei getrennte Ansichten: Designer und Blocks. Der Designer dient der Gestaltung des User Interface und der Auswahl aller Komponenten (Buttons, Grafiken, Töne, Timer, GPS). Darauf aufbauend stehen im Blocks Editor allgemeine und komponentenspezifische Blöcke (abhängig von den eingefügten Elementen im Designer) zur Programmierung der Funktionalität zur Verfügung. Eine Besonderheit bei der Arbeit mit dem App Inventor ist das Live-Testing, welches über einen integrierten Emulator oder auch direkt auf einem Android-Smartphone bzw. -Tablet möglich ist. Die von den Lernenden entwickelten Apps können somit in die Realität (auf das Smartphone) überführt werden und sind nicht auf eine künstliche Mikrowelt im Browser beschränkt. Dieser Unterschied muss bei der späteren Auswertung insbesondere hinsichtlich des Bildes der Disziplin berücksichtigt werden.

3.2 Beschreibung der Schülerlabor-Module

Nachdem die beiden Werkzeuge beschrieben und verglichen sind, werden hier die konkreten Workshops im Rahmen des InfoSphere vorgestellt, anhand derer die Effekte grafischer Programmierwerkzeuge evaluiert wurden.

Das Modul zu Scratch nennt sich „*Spielend Programmieren lernen mit Scratch*“³ und wird für Klassen der Unter- und Mittelstufe angeboten. Das didaktische Konzept hinter diesem Workshop ist ein Gruppenpuzzle. Dies bedeutet, dass jeweils vier Kinder ein Team bilden, dessen Ziel es ist, am Ende des Tages gemeinsam ein Spiel zu programmieren. Noch vor dem Start wird allen Teilnehmerinnen und Teilnehmern gemeinsam die Webseite Scratch präsentiert und gezeigt, wie die Bausteine zusammengesetzt werden, ohne vertieft auf die Bedeutung einzelner Befehle einzugehen. In der ersten Phase des Gruppenpuzzles teilen sich die Teams auf, sodass jedes Teammitglied gemeinsam mit Mitgliedern anderer Teams Experte auf einem bestimmten Gebiet wird. Dabei entstehen Expertengruppen für die Themenbereiche „Bewegung und Zeichnen“, „Fühlen und Variablen“, „Grafik und Aussehen“ und „Klang und Kommunikation“. In diesen Expertenteams lernen sie mit Hilfe kleinschrittiger Anleitungen und anhand konkreter Beispiele eine Auswahl der Befehle kennen. In der zweiten Phase des Gruppenpuzzles entwickeln die vier Teammitglieder gemeinsam ein Spiel, zu dem die grobe Spielidee und mögliche Erweiterungsansätze gegeben sind. Dabei trägt jedes Teammitglied eine Verantwortung, da jede Expertin bzw. jeder Experte für ein bestimmtes Thema zuständig ist, welches für die Umsetzung des Spiels benötigt wird. In dieser Phase können die Kinder sehr kreativ arbeiten, da Scratch zahlreiche grafische und auch akustische Optionen zur Ausgestaltung des Spiels bietet. Zum Abschluss präsentieren alle Teams ihre Spiele vor der Gruppe und können diese anschließend gegenseitig austesten. Darüber hinaus ist es möglich, dass die Teilnehmerinnen und Teilnehmer ihre Spiele zur weiteren Bearbeitung mitnehmen.

Zum App Inventor gibt es im Schülerlabor InfoSphere zwei unterschiedliche Module. Das Modul „*Erste App programmieren*“⁴ ist als Einstieg für Klassen der Stufen 6 bis 8 konzipiert, wohingegen das Modul „*InfoSphere goes Android*“⁵ als Fortsetzung oder für Anfänger der Stufen 8 bis 10 angeboten wird. Der Unterschied liegt im Detailgrad der Anleitungen wie auch in der Übersetzung aller englischen Begriffe. Die Sozialform in beiden Modulen ist die Partnerarbeit an einem Laptop und mit einem Smartphone. Auch hier starten beide Module mit einer kurzen Präsentation des Werkzeugs.

Im Modul „*Erste App programmieren*“ wird in drei Schritten eine Zeichen-App mit Foto-Funktion entwickelt. Mittels des Arbeitsblattes „Zeichnen für Anfänger“ wird programmiert, wie ein Foto aufgenommen und als Hintergrund gespeichert werden kann. Im Blatt „Zeichnen für Fortgeschrittene“ wird die App um Buttons erweitert, so dass die Stiftfarbe aus den vier Grundfarben gewählt werden kann und auch bereits Punkte und Linien in unterschiedlichen Strichstärken gezeichnet werden können. In dem optionalen Zusatzblatt „Zeichnen für Profis“ wird die App um eine komplette Farbpalette erweitert.

Das Modul „*InfoSphere goes Android*“ hingegen startet nach der Einführung mit einer ersten Phase, in der alle Zweierteams eine einfache „Hallo InfoSphere“-App entwickeln,

³ Modulbeschreibung „Scratch“: <http://schuelerlabor.informatik.rwth-aachen.de/module/scratch>

⁴ Modulbeschreibung „Erste App“: <http://schuelerlabor.informatik.rwth-aachen.de/module/erste-app>

⁵ Modulbeschreibung „InfoSphere goes Android“: <http://schuelerlabor.informatik.rwth-aachen.de/module/is-goes-android>

die kleinschrittig angeleitet wird und den Umgang mit dem App Inventor übt. Anschließend hat jedes Team die Wahl zwischen den vier Spielideen „AngryBlob“, „Asteroids“, „FlappyBird“ und „MoleMash“, welche alle an existierende Apps angelehnt und daher den Schülerinnen und Schülern unter Umständen bereits bekannt sind. Hierbei wird der Schwierigkeitsgrad der Apps auf den Anleitungen angegeben, so dass die Teams nach der „Hallo InfoSphere“-App diesen selbst auswählen können. Die Anleitungen sind hier weniger kleinschrittig, so dass den Lernenden mehr Verantwortung über den eigenen Lernprozess übertragen wird. Auch haben die Lernenden die Möglichkeit ihre App nach eigenen Ideen weiterzuentwickeln.

4 Auswertung der Evaluation

In den Jahren 2015 und 2016 wurden für die vorgestellten drei Module zum grafischen Programmierereinstieg 694 Pre- und 720 Posttestdatensätze erhoben. Da die Vorbefragung in der Regel im Vorfeld in der Schule ausgefüllt wird, kommt es vor, dass Kinder (z. B. aufgrund von Krankheit) nicht an beiden Erhebungszeitpunkten einen Fragebogen ausfüllen. Aus dieser Problematik heraus ergibt sich, dass nur 564 Pre-Post-Datensätze korrekt zugeordnet werden konnten. Da zur Beforschung der fünf Argumente Informatikeinsteiger im Fokus stehen, wurde die Stichprobe für die Auswertung auf Schülerinnen und Schüler der Klassenstufen 6 bis 9 eingeschränkt. Dies ergibt einen Stichprobenumfang von $N = 471$ (siehe auch Abb. 2).

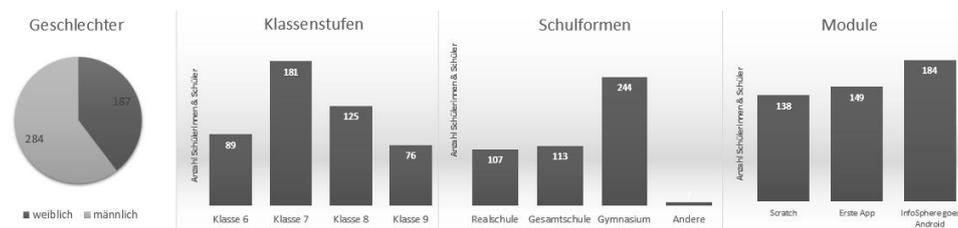


Abb. 2: Stichprobenbeschreibung

Argument 1 - Begeisterungsfähigkeit

Im Schnitt werden alle drei Module von den Teilnehmerinnen und Teilnehmern positiv bewertet (siehe Abb. 3 a). Die Antworten hängen dabei signifikant vom Geschlecht der Kinder ($p < .05$), jedoch nicht vom Alter, der Klassenstufe und dem besuchten Modul ab. Die Daten der Mädchen zeigen mit einem Mittelwert von $M=2,44$ ($s=1,380$), dass den Mädchen das Modul signifikant besser gefallen hat als den Jungen mit $M=2,53$ ($s=1,598$)⁶. Obwohl die Differenzen nicht signifikant sind, zeigt ein Vergleich der Mittelwerte, dass das Modul „Scratch“ am beliebtesten und „InfoSphere goes Android“ am unbeliebtesten ist, was möglicherweise mit dem unterschiedlichen

⁶ Die Kodierung geschieht mittels „trifft voll zu“ (1) bis „trifft gar nicht zu“ (5).

Schwierigkeitsgrad der Module oder der tendenziell älteren Zielgruppe zusammenhängen könnte.



Abb. 3 Auswertung bezüglich Begeisterungsfähigkeit

Bezüglich der Aussage, ob sie sich in Zukunft verstärkt mit Themen der Informatik beschäftigen möchten, ergibt sich die Verteilung aus Abb. 3 b). Auch die Antwort auf diese Frage ist wieder signifikant vom Geschlecht abhängig ($p < .01$). Entgegen der obigen Aussage wollen sich hierbei die Jungen mit $M=2,87$ ($s=1,326$) eher zukünftig mit der Informatik beschäftigen als die Mädchen mit $M=3,12$ ($s=1,135$). Hier wird deutlich wie stark der, im Vorfeld generierte, Wunsch der Kinder bereits ausgeprägt ist und wie begrenzt dabei eine punktuelle Maßnahme wirkt. Darüber hinaus ist die Bewertung dieser Aussage signifikant abhängig von der Klassenstufe ($p < .05$). Schülerinnen und Schüler niedriger Klassenstufen geben deutlicher an, sich verstärkt mit Informatik beschäftigen zu wollen. Dies könnte in den geringeren Vorerfahrungen der jungen Zielgruppe begründet sein, so dass diese erst durch das Modul einen Einblick in die Informatik erhalten und durch ihre Erfolgserlebnisse bestärkt wurden.

Als Antwort auf die dritte Aussage „Ich würde gerne an weiteren Modulen des InfoSphere teilnehmen.“ ergibt sich eine nahezu Gleichverteilung zwischen den fünf Antwortalternativen (siehe Abb. 3 c)). Auch hier konnten wieder geschlechtsspezifische Unterschiede festgestellt werden ($p < .01$). Obwohl ihnen der Modulbesuch weniger gut gefallen hat, haben Jungen ($M=2,88$, $s=1,397$) ein signifikant größeres Interesse am Besuch weiterer Module im Schülerlabor als Mädchen ($M=2,99$, $s=1,255$). Auch hier zeigt sich wieder wie gefestigt der vorher ausgeprägte Wunsch einer möglichen Beschäftigung mit der Informatik ist und wie eingeschränkt ein eintägiges Modul dahingehend wirken kann.

Argument 2 - Vermittlung eines interessanten Bildes der Disziplin

Als erstes wurde hier die persönliche Einschätzung zwischen den beiden Extrema „Informatik ist interessant (1) versus uninteressant (100).“ erhoben. Dabei zeigen sich signifikante Differenzen zwischen den Angaben vor und nach dem Modulbesuch ($p < .05$). Dabei liegt der Mittelwert vorher bei $M=24,98$ ($s=26,427$) und nachher bei $M=27,29$ ($s=27,661$). Aus dieser Verschiebung lässt sich ablesen, dass die Informatik im Schnitt nach dem Modul leicht „uninteressanter“ wahrgenommen wird. Spannend sind die signifikanten Geschlechterunterschiede ($p < .05$). Bei Jungen hat sich die Bewertung

um 3,90 Punkte Richtung „uninteressant“ verschoben, wohingegen sie sich bei Mädchen um 0,10 Punkte Richtung „interessant“ verändert hat, was unter anderem darin zu begründen ist, dass das Interesse von den Jungen im Vorfeld bereits extrem hoch angegeben wurde (Jungen vorher $M=19,11$, Mädchen vorher $M=33,89$). Obwohl dies nicht beabsichtigt war, kann der Rückgang des Interesses bedeuten, dass das vorherige Bild der Disziplin positiv verzerrt war und nun ein realistisches Bild entstanden ist.

Bei der zweiten Aussage ging es darum, ob die Informatik eher langweilig (1) oder spannend (100) ist. Für die gesamte Stichprobe konnten keine signifikanten Veränderungen durch die Module bemessen werden. Allerdings zeigte sich, dass die Antworten signifikant vom Geschlecht der Befragten abhängen ($p<.05$). Bei Jungen veränderte sich der Mittelwert um 2,33 Punkte Richtung „langweilig“, bei Mädchen entgegengesetzt um 2,29 Punkte Richtung „spannend“. Dabei gaben die Jungen im Pretest wesentlich positivere Werte ($M=77,51$) an als die Mädchen ($M=60,25$).

Auch bezüglich der Aussage „Informatik ist eintönig (1) versus abwechslungsreich (100).“ ergaben sich insgesamt keine signifikanten Differenzen zwischen den beiden Erhebungszeiträumen. Allerdings zeigte sich hier eine signifikante Abhängigkeit vom besuchten Modul. Durch den Besuch des Moduls „Scratch“ veränderte sich der Mittelwert um 2,19 Punkte in Richtung „abwechslungsreich“, nach dem Modul „Erste App“ um 0,39 Punkte ebenfalls in Richtung „abwechslungsreich“, nach dem Modul „InfoSphere goes Android“ jedoch um 0,29 Punkte in Richtung „eintönig“.

Argument 3 - Passung zu den Erwartungen der Programmierneinsteiger

Bezüglich dieses dritten Argumentes sollte die Zustimmung zu der expliziten Aussage „Das Modul entsprach meinen Erwartungen.“ angegeben werden. Hierbei ergab sich folgende Verteilung in der gesamten Stichprobe: 115x „trifft voll zu“ (1), 102x „trifft zu“ (2), 130x „weder noch“ (3), 70x „trifft eher nicht zu“ (4), 54x „trifft gar nicht zu“ (5). Auch hier ergaben sich wieder signifikante Geschlechterunterschiede ($p<.05$). Dabei wurde die Erwartung der Mädchen mit $M=2,59$ ($s=1,180$) stärker erfüllt als diese der Jungen $M=2,73$ ($s=1,377$).

Weiter wurde analysiert, welche Erwartungen diejenigen mit ins Schülerlabor brachten, welche die deutlichsten Verschiebungen hin zu „Informatik ist uninteressant.“, „Informatik ist langweilig.“ und „Informatik ist eintönig“ verzeichneten. Die Erwartungen dabei entsprechen inhaltlich dem besuchten Modul, also beinhalteten z. B. „App-Programmierung“, „etwas Neues lernen“ und „Spaß haben“. Eine Vermutung für die negative Interessenentwicklung könnte sein, dass diejenigen enttäuscht wurden, die erwarteten, dass ihnen der Freiraum einer komplett eigenen Idee für die App bzw. das Spiel gelassen würde, was mehrfach in den Durchführungen als Kritik geäußert wurde.

Die Auswertung der Verbesserungsvorschläge ist leider wenig zielführend, da meist „keine“ oder unklare Angaben wie „interessanter gestalten“ genannt wurden.

Argument 4 - Einfluss auf die wahrgenommene Schwierigkeit der Disziplin

Mittels der im Pre- sowie im Posttest zu wertenden Aussage „Informatik ist schwierig (1) versus einfach (100).“ wird das vierte Argument beleuchtet. Hier zeigt sich eine signifikante Veränderung zwischen den beiden Erhebungszeitpunkten ($p < .05$). Die Informatik wird nach dem Modul mit $M=48,2145$ ($s=24,91$) schwieriger eingeschätzt als im Vorfeld $M=51,3316$ ($s=25,00$). Auch wenn es keine signifikanten Unterschiede zwischen den Teilgruppen gibt, zeigt ein Blick in die Daten, dass diese Veränderung bei Mädchen deutlicher stattfindet. Dieses Ergebnis bestärkt den Verdacht, dass einige Jugendliche mit einem positiv verzerrten Bild der Informatik das Schülerlabor besuchen und von den Herausforderungen der Programmierung überrascht wurden.

Argument 5 - Vermittlung eines kreativen Berufsbildes „Informatiker-in“

Mit Blick auf das fünfte Argument wird die Zustimmung zu zwei Aussagen auf einer Skala von „stimme gar nicht zu“ (1) bis „stimme voll zu“ (11) erhoben. Dabei zeigt sich ein signifikanter Unterschied bei der Aussage „Informatiker-innen müssen kreativ sein.“ ($p < .05$). Nach dem Modul wird dieser Aussage mit $M=7,9427$ ($s=2,66$) signifikant stärker zugestimmt als vorher $M=7,4268$ ($s=3,05$). Mittels eines grafischen Programmereinstiegs gelingt es den Schülerinnen und Schülern zu verdeutlichen, dass Softwareentwicklung ein kreativer Prozess sein kann. Dabei sind die Veränderungen von keiner trennenden Variablen signifikant abhängig. Auch bei der Auswertung der Pre-Test-Daten zeigten sich keine spezifischen Unterschiede unter den Teilgruppen.

Zum Abschluss wurde die Aussage „Informatiker-innen kreieren viel Neues.“ evaluiert. Hierbei zeigten sich im Vorfeld wie auch im Vergleich zwischen den beiden Erhebungszeiträumen keine signifikanten Unterschiede.

5 Reflexion & Ausblick

Insgesamt hat die quantitative Auswertung der 471 Pre-Posttest-Datensätze gezeigt, dass die Module im Schülerlabor zum Einstieg in die grafische Programmierung mittels Scratch und dem App Inventor nicht unbedingt zu einem positiveren Bild der Informatik beitragen, aber vermutlich ein realistischeres Bild erzeugen.

Argument 1 - Begeisterungsfähigkeit: Die Auswertung zeigt, dass die Module den Mädchen signifikant besser gefallen haben, dennoch sind es die Jungen, die angeben sich in Zukunft eher mit Informatik beschäftigen zu wollen. Weiter zeigt sich, dass das anspruchsvollste Modul am schlechtesten bewertet wird, was ein Indiz dafür ist, dass Überforderung zur Abnahme der Begeisterung führt.

Argument 2 - Vermittlung eines interessanten Bildes der Disziplin: Bei einem Teil der Jungen, veränderte sich das Bild der Informatik teils negativ. Dieser Rückgang des Interesses kann auch in einem im Vorfeld extrem positiven, unter Umständen gar verzerrten Bildes der Informatik begründet sein. Das Bild der Informatik veränderte sich bei Jungen in Richtung „langweilig“, bei Mädchen jedoch exakt entgegengesetzt in

Richtung „spannend“. Somit kann davon ausgegangen werden, dass sich die eigene Vorstellung mancher Besucherinnen und Besucher hin zu einer realistischeren Einschätzung entwickelt hat. Weiter haben sich die zuvor stark ausgeprägten Unterschiede zwischen den Geschlechtern leicht angeglichen.

Argument 3 - Passung zu den Erwartungen der Programmierneinsteiger: Die Auswertung des Pretests zeigte, dass die Erwartungen zu den Inhalten der Module passen, es jedoch Enttäuschungen bezüglich der gewünschten Freiheitsgrade innerhalb der Projekte gab. Insgesamt gaben Mädchen deutlicher als Jungen an, dass die Module ihre Erwartungen erfüllt hätten. Dies könnte eine Begründung für die negativen Verschiebungen bzgl. der wahrgenommenen Interessantheit der Disziplin bei den Jungen sein.

Argument 4 - Einfluss auf die wahrgenommene Schwierigkeit der Disziplin: Die Tatsache, dass die Informatik nach einem Programmierneinstieg als schwieriger bewertet wird, legt nahe, dass einige Lernende mit einem verzerrten Bild der Informatik das Schülerlabor besuchten und von den Anforderungen überrascht wurden.

Argument 5 - Vermittlung eines kreativen Berufsbildes „Informatiker-in“: Der grafische Programmierneinstieg zeigt die Softwareentwicklung als kreativen Prozess und führt zur signifikanten Überzeugung, dass Informatiker-innen kreativ sein müssen.

Da der Einstieg in die Programmierung dank passender Werkzeuge zukünftig bereits im Grundschulalter stattfinden kann, soll die Evaluation auf diese jüngere Zielgruppe ausgeweitet werden. Dazu werden auch Apps zum grafischen Programmierneinstieg wie „Scratch Junior“ und Hardware wie LEGO Wedo, Ozobot und Calliope hinsichtlich ihres Potentials bezüglich der Schülervorstellungen über Informatik beleuchtet.

Literaturverzeichnis

- [Be15] Bergner, N.: Konzeption eines Informatik-Schülerlabors und Erforschung dessen Effekte auf das Bild der Informatik bei Kindern und Jugendlichen. Dissertation, Aachen, 2015.
- [Ma10] Maloney, J. et al.: The Scratch Programming Language and Environment. In ACM Transactions on Computing Education, 2010, 10; 16: 1–15.
- [Ut10] Utting, I. et al.: Alice, Greenfoot, and Scratch -- A Discussion. In ACM Transactions on Computing Education, 2010, 10; S. 1–11.
- [Wo11] Wolber, D.: App inventor and real-world motivation. In (Cortina, T. J. Hrsg.): Proceedings of the 42nd ACM technical symposium on Computer science education. ACM, New York, NY, 2011; S. 601.