Niels Pinkwart, Johannes Konert (Hrsg.): Die 17. Fachtagung Bildungstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 235

PLAY & CODE – Eine Serious Game basierte Lernplattform

Elias John[1]

Abstract: Viele Positionen im Informatiksektor können schon heute nicht besetzt werden und es ist davon auszugehen, dass dieser Mangel an Informatiker*innen in Zukunft noch steigen wird. Problematisch ist dabei auch die hohe Abbrecherquote in den entsprechenden Studiengängen. Viele Themen des Studiums erscheinen als zu theoretisch, abstrakt oder nicht praxisrelevant. Es stellt sich daher die Frage, wie Lerninhalte adäquater vermittelt und mehr Schüler*innen für Themen der Informatik begeistert werden können. Als Mittel zur Steigerung der Motivation scheinen besonders Spiele geeignet. Der vorliegende Aufsatz soll daher folgende Frage beantworten: Können mithilfe eines Serious Games basierten Lernsystems Motivation und Lernerfolg für grundlegende Programmierkenntnisse gesteigert werden?

Keywords: Serious Games, Grundlagen der Programmierung, JavaScript, Lernplattform

1 Einleitung

Durch die fortschreitende Digitalisierung besteht ein hoher Bedarf an Informatiker*innen. Trotz einer zunehmenden Anzahl an Informatik-Studierenden steigt die Zahl der unbesetzten Stellen seit 2014 pro Jahr stetig an [IW17]. Dieser Mangel an Fachkräften wird zudem durch eine hohe Abbrecherquote im Studium verstärkt [He17]. Nach Muratet et al. [Mu09] empfinden viele Studierende zu Beginn des Studiums das Themengebiet als zu theoretisch und abstrakt. Wie können also Lerninhalte praxisnäher vermittelt sowie bereits Schüler*innen für Themen der Informatik begeistert werden? Als Mittel für die Steigerung der Motivation scheinen besonders Spiele geeignet, da sie einen leichten und niedrigschwelligen Zugang zum jeweiligen Thema ermöglichen. Es stellt sich damit die Frage, ob mithilfe eines Serious Games basierten Lernsystems Motivation und Lernerfolg für grundlegende Programmierkenntnisse gesteigert werden können.

Dafür wird im folgenden Kapitel 2 zunächst auf den Stand der Forschung eingegangen. Bestehende Serious Games und ihre Eigenschaften werden vorgestellt. Anschließend wird im dritten Kapitel näher auf das Konzept der Serious Game basierten Lernplattform PLAY & CODE eingegangen. Diese vermittelt grundlegende Programmierkonstrukte mittels Serious Games und passt die Texte individuell an den/die Benutzer*in an. Um die Forschungsfrage zu beantworten, wurde die Plattform an mehreren Schulen evaluiert. Die

[1] Humboldt-Universität zu Berlin, Institut für Informatik, Rudower Chaussee 25, 12489 Berlin, elias.john@student.hu-berlin.de

doi:10.18420/delfi2019_248

Methodik und Ergebnisse der Evaluation werden im vierten Kapitel vorgestellt. Abschließend wird die Forschungsfrage beantwortet und ein Fazit gezogen.

2 Stand der Forschung

Für ein erfolgreiches Lernen ist die intrinsische Motivation des/der Lernenden eine wichtige Voraussetzung [Hu07]. Als ein erfolgsversprechendes Mittel zur Steigerung der Motivation gelten Spiele [Dö16]. Solche Serious Games werden in verschiedenen Bereichen, wie beispielsweise der Biologie und Chemie, eingesetzt (vgl. [Ch15]). Exemplarisch werden einige solcher Serious Games aus dem Bereich der Informatik vorgestellt. Dabei lässt sich feststellen, dass die Programmierung auf verschiedene Weise (textuell oder grafisch) vermittelt und in verschiedenen Umgebungen eingebettet werden kann (unterrichtsbegleitend, alleinstehend, sowie als eigenständiges Spiel oder eingebettet in einer Lernumgebung).

Das Spiel **„Program your Robot“** von Kazimoglu et al. [Ka12] ist ein Serious Game zum Üben grundlegender Programmierkonstrukte, wie konditionaler Logik und Schleifen. Den Lernenden wird dabei vor allem vermittelt, wie Algorithmen aufgebaut und Programme systematisch auf Fehler untersucht werden können. Ziel des Spiels ist es, einem Roboter bei der Flucht zu helfen. Dafür muss der/die Spieler*in einen Lösungsalgorithmus entwickeln, indem er vorgefertigte grafische Befehlssymbole in beliebiger Reihenfolge in eine begrenzte Anzahl von Slots zieht. Zufällig erzeugte Belohnungsitems steigern den Punktestand der Spieler*in.

„Prog&Play“ ist ein Serious Game von Muratet et al. [Mu09] zur Übung von Programmierung. Das Serious Game ist begleitend zum Unterricht durch eine Lehrkraft gedacht und enthält keine Lehrinhalte. Das Spielgeschehen findet im Inneren eines gehackten Computersystems statt, über welches der/die Spieler*in in mehreren Leveln mit dem Steuern von Elementen wie Bits und Bytes wieder die Kontrolle erlangen muss. Die Lösung für das jeweilige Problem wird mittels einer zur Verfügung gestellten API programmiert, die durch eine Vielzahl an Programmiersprachen angesteuert werden kann.

Schäfer et al. [Sc13] stellen eine **kollaborative, Multitouch Lernumgebung** vor. Ziel ist es, gemeinsam an einem Multitouch-Gerät mathematische Logik zu erlernen und zu üben. Nach den Autor*innen führt Kollaboration zu mehr Interaktion und einer besseren Aufnahme des Lernstoffes. Neue Lerninhalte werden zunächst grafisch animiert erklärt. Anschließend werden Aufgaben von dem/der Benutzer*in selbst ausgeführt. Dabei gibt das System Feedback und stellt Quizfragen zum Thema. Im Freeplay-Abschnitt können weitere Aufgaben gelöst werden. Hierbei wird der/die Benutzer*in mit Punkten für schnelle Reaktionen belohnt, welche für andere Spielende einsehbar sind. Dadurch sollen sich andere Spieler*innen motiviert fühlen, bessere Werte zu erreichen.

Von den genannten Arbeiten gehen nur Schäfer et al. [Sc13] näher auf die verwendeten didaktischen Modelle und Theorien ein. Dies scheint ein generelles Phänomen zu sein: In

53 untersuchten Veröffentlichungen zu Serious Games von 2002 bis 2013 stellten Cheng et al. [Ch15] fest, dass nur die Hälfte der Autor*innen den Einsatz solcher Spiele lerntheoretisch begründen. Zudem wurde die Effizienz von Serious Games empirisch noch nicht vollständig nachgewiesen: So stellen beispielsweise Girard et al. [GEM13] in ihrer Meta-Analyse zu Serious Games diverse methodische Mängel bei der Untersuchung der betrachteten Artikel fest. Serious Games scheinen ein hilfreiches Werkzeug für Lernzwecke zu sein, jedoch haben bisherige Evaluationsstudien noch keine hinreichende Begründung für den Erfolg des Einsatzes von Serious Games geliefert. Im Folgenden wird daher eine didaktisch fundierte Serious Game basierte Plattform zum Erlernen von Programmierung vorgestellt.

3 Konzept der Lernplattform PLAY & CODE

3.1 Didaktik der Lernplattform

Didaktisch gründet die im folgenden vorgestellte Lernplattform auf dem ARCS-Modell von Keller [Ke87] sowie den „First Principles of Instruction“ von Merrill [Me02]. Das Modell von Merrill basiert auf einer umfangreichen Untersuchung verschiedener Instruktionsmodelle und versucht die Ansätze in einem Modell zu vereinen. Die Arbeit Merrills fußt auf dem problembasierten Lernansatz, welcher sich als sehr effektiv erwiesen hat (vgl. [Dö16] und [Sa06]). Durch den Problembezug wird der Lerninhalt konkretisiert und wirkt so weniger abstrakt, was sich unter anderem positiv auf die Motivation der Lernenden auswirken kann. Zusätzlich wird das ARCS-Modell von Keller berücksichtigt, welches verschiedene Aspekte zur Motivationserzeugung beschreibt. Der Einsatz eines Serious Game als Lehrmittel trägt zum Spaß beim Lernen bei und ruft intrinsische Motivation hervor. Da Programmieraufgaben auf verschiedene Wege gelöst werden können, wird außerdem Kreativität beim Lernen befördert.

Im Gegensatz zu anderen spielebasierten Lernplattformen, wie „Program your Robot“ von Kazimoglu et al. [Ka12], wird hier keine grafische, sondern die textuelle Programmiersprache JavaScript eingesetzt. Grafische Programmiersprachen scheinen einen sehr leichten Zugang zur Programmierung zu erlauben. Im Hinblick auf die von Merrill [Me02] und Keller [Ke87] geforderte Realitätsnähe und Alltagstauglichkeit wurde bei PLAY & CODE jedoch nach einer Sprache gesucht, welche im professionellen Bereich eine starke Verbreitung findet. Hierfür scheint sich JavaScript besonders zu eignen. So listet beispielsweise der TIOBE Programming Community Index [TI19] JavaScript unter den Top zehn Programmiersprachen. Die Schüler*innen können ihr erlerntes JavaScript-Wissen nach dem Lernkurs direkt und vielseitig in Web-, Desktop- und Serveranwendungen verwenden.

3.2 Struktur des Systems

Die Lernplattform ist als ein Online-Mehrbenutzersystem konzipiert, bei der sich Interessierte selbstständig anmelden können.[2] Nach dem erstmaligen Login muss der/die Benutzer*in zunächst einen kurzen Fragebogen zu Programmier-Vorerfahrungen beantworten. Dabei wird mit Likert-Skalen von 1 (keine Erfahrung) bis 5 (sehr erfahren) nach Programmiererfahrungen im Allgemeinen und Erfahrungen mit JavaScript im Speziellen gefragt. Diese Informationen werden zur Errechnung eines Erfahrungswertes verwendet, welcher die Länge der Lehrtexte beeinflusst. Hierauf wird später detaillierter eingegangen.

Der/Die Benutzer*in kann aus einer Kursübersicht seine/ihre Kurse frei auswählen, welche mit einem kurzen Text und einem Thumbnail beschrieben sind. Hier kann ein Kurs entweder begonnen oder zu einem späteren Zeitpunkt auch fortgesetzt werden. Jeder Kurs ist in mehrere Lektionen gegliedert, welche der Reihe nach abgeschlossen werden müssen.

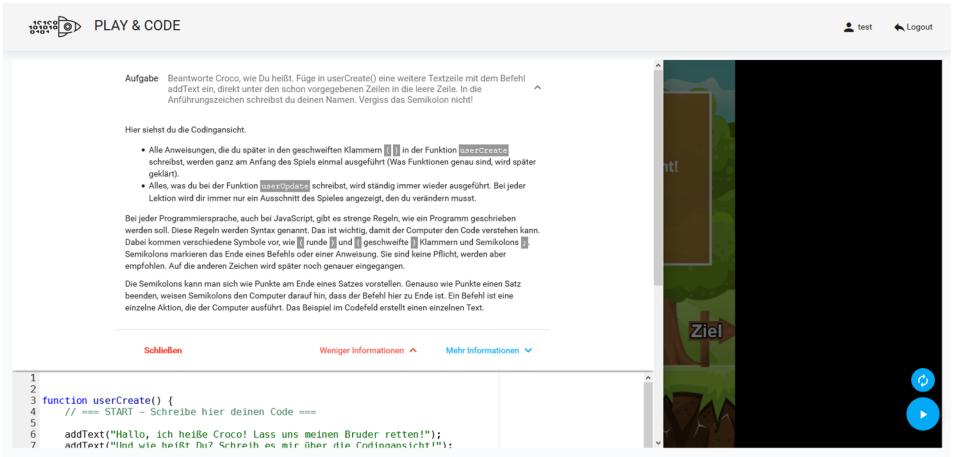

Abb. 1: Screenshot aus dem Kurs "Crocos Bruder" mit geöffneter Coding-Ansicht.

Jeder Kurs stellt ein Videospiel dar, mit welchem eine in sich abgeschlossene Geschichte erzählt wird. Die Lektionen des Kurses sind die einzelnen, aufeinanderfolgenden Abschnitte dieses Spiels. In jeder Lektion wird in einer animierten Intro-Sequenz die Geschichte weitererzählt und der Protagonist vor ein neues Problem gestellt. Hiermit soll sich der/die Benutzer*in in das Spielgeschehen integriert fühlen. Auf die Intro-Sequenz folgt die Lern- und Programmierphase, in welcher der/die Benutzer*in neue Inhalte erlernt. Dafür muss zunächst die Coding-Ansicht geöffnet werden, in welcher die Lehrtexte, die Aufgabe und die Codeeingabe angezeigt werden. Die Texte können auch Bilder und formatierte Texte enthalten (vgl. Abb. 1). Der Text unterteilt sich in folgende Absätze:

[2] Implementiert wurde die Lernplattform PLAY & CODE mit dem Java-Framework Spring Boot für das Backend und mit dem Frontend-Framework Angular 6. Das Backend verwaltet dabei alle relevanten Daten in einer MySQL-Datenbank und stellt diese über eine REST-API für das Frontend bereit. Dabei werden Abfragen authentifiziert und autorisiert.

1.: Grundlegende Informationen zum jeweiligen Lernthema werden bereitgestellt. 2.: Informationen werden weiter ausgeführt und Konstrukte detaillierter erklärt. 3.: Konkrete Positiv- und Negativbeispiele werden aufgelistet und weiterführende Ressourcen verlinkt.

3.3 Adaptivität der Lehrtexte

Da jeder/jede Lernende einen eigenen Lernstil und unterschiedliche Voraussetzungen mitbringt, ist es notwendig, auf diese individuell einzugehen. Diverse Studien haben nach Vandewaetere et al. [VDC11] gezeigt, dass dieser individuumsbasierte Ansatz uniformen Modellen, die die Bedürfnisse der Lernenden kaum berücksichtigen, überlegen ist. Der Lehrtext wird daher an den/die Benutzer*in adaptiert: Je nachdem, wie hoch der errechnete Erfahrungswert der Benutzer*innen ist, wird ein unterschiedlich großer Ausschnitt des gesamten Textes angezeigt. Der jeweilige Erfahrungswert der Benutzer*innen ist auf einer Skala von 1 bis 15 angesiedelt. Bei einem Wert unter 6 werden alle drei Abschnitte angezeigt: Der/die Benutzer*in benötigt detaillierte Informationen und konkrete Beispiele, um die Aufgabe lösen zu können. Insbesondere Positiv- und Negativbeispiele sind nach Merrill [Me02] hilfreich, Inhalte zu veranschaulichen. Bei einem Erfahrungswert zwischen 6 bis 10 werden zwei Abschnitte angezeigt. Bei Werten darüber wird lediglich der erste Abschnitt dargestellt: Der/Die Benutzer*in kennt bereits die Paradigmen aus anderen Sprachen und benötigt lediglich knappe Informationen über die JavaScript spezifische Verwendung. Meint der/die Benutzer*in mehr oder weniger Informationen zu benötigen, kann er/sie den Text weiter ein- bzw. ausklappen. Der Erfahrungswert errechnet sich aus den Informationen des Fragebogens, den benötigten Versuchen in der aktuellen Lektion und den manuellen Veränderungen des Textausschnitts. Der Wert dient dabei lediglich zur Berechnung und wird dem/der Benutzer*in nicht angezeigt. Mit diesem mikroadaptiven Ansatz[3] (vgl. [VDC11]) soll Benutzer*innen mit Vorerfahrungen zu viel Text erspart bleiben und Anfänger*innen die benötigten Informationen zur Verfügung gestellt werden.

3.4 Spielablauf

Nachdem Lehrtexte und Aufgabenstellung von dem/der Benutzer*in gelesen wurden, kann er/sie mit dem neu gewonnenen Wissen eine Lösung für das Problem programmieren. Der für die Spielmechanik benötigte Code wird dabei in – für den/die Benutzer*in nicht sichtbare – Dateien ausgelagert. Die Benutzer*innen sehen lediglich die für sie bestimmte Datei UserCode, welche befüllt werden muss, um die Aufgabe zu lösen. Damit soll sichergestellt werden, dass der/die Benutzer*in nicht von der Codefülle überfordert ist, sondern sich auf das Wesentliche konzentrieren kann. Die Datei UserCode enthält initial zwei Funktionen: userCreate und userUpdate. userCreate wird einmalig bei der

[3] Im Gegensatz zu makroadaptiven Ansätzen, bei denen der/die Lernende einmalig klassifiziert und Inhalte darauf basierend bereitgestellt werden, werden die Inhalte beim mikroadaptiven Ansatz dynamisch während der Kursdurchführung angepasst [VDC11]. Bei Letztem werden also individuelle, möglicherweise temporäre Eigenschaften des/der Lernenden stärker berücksichtigt und damit auf Mikroebene agiert.

Initialisierung des Spiellevels ausgeführt, während userUpdate fortlaufend in der Spielschleife einmal pro Bilddarstellung (frame) aufgerufen wird. Die Funktionen können leer sein oder vorgegebene Codes enthalten, die von dem/der Benutzer*in ergänzt werden müssen.

Nachdem der/die Benutzer*in seinen/ihren Lösungscode geschrieben hat, kann er/sie auf „Play“ drücken. Sollten syntaktische Fehler, wie zum Beispiel falsch gesetzte Klammern, vorhanden sein, wird die Ausführung unterbrochen und die Fehler werden mit Zeilennummern und Fehlermeldung angezeigt. Andernfalls wird das modifizierte Level gestartet. Dies entspricht der Empfehlung Merrills [Me02], dass der/die Benutzer*in erworbenes Wissen unmittelbar anwenden kann und Feedback erhält. Der/Die Benutzer*in kann dabei das Problem auf verschiedene, kreative Weise lösen, solange die semantischen Anforderungen erfüllt sind. Diese werden vom Kursautor oder von der Kursautorin definiert. Einerseits können Werte und Zustände, wie die Position der Spielfigur, überprüft werden. Andererseits kann auch die Codeabgabe auf bestimmte Schlagwörter untersucht werden.

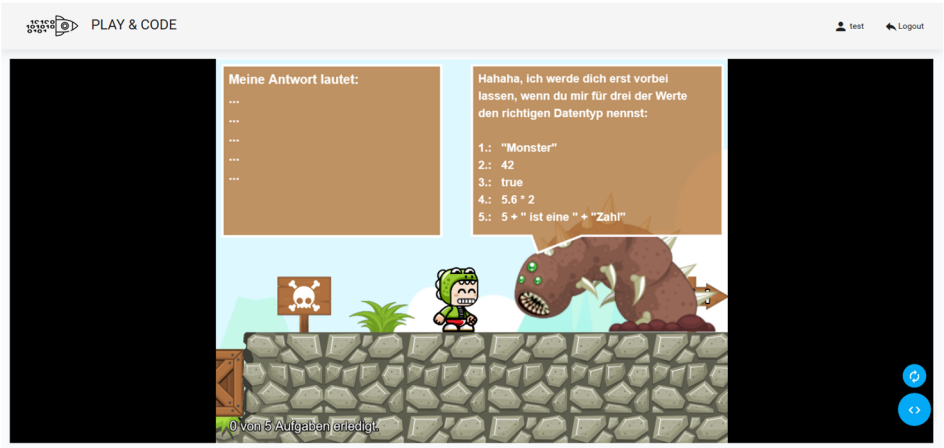

Abb. 2: Screenshot aus dem dritten Level von "Crocos Bruder"

Durch das Lösen weiterer Probleme durch zusätzliche, optionale Programmierung erscheinen in einigen Leveln Diamanten, welche der/die Benutzer*in einsammeln kann. Je nachdem ob und wie viele Diamanten eingesammelt wurden, wird dem/der Benutzer*in nach Levelabschluss eine Bronze-, Silber oder Goldmedaille vergeben. Dadurch soll der Erfolg sichtbar gemacht und eine Zufriedenheit der Benutzer*innen erreicht werden (vgl. Kellers Satisfaction [Ke87]). Im unteren Bildschirmbereich wird ihm/ihr die Gesamtanzahl der zu lösenden Teilprobleme und der aktuelle Fortschritt angezeigt (siehe Abb. 2). In Leveln ohne Diamanten werden Goldmedaillen vergeben. Im Benutzerprofil können gewonnene Trophäen betrachtet werden. Zusätzlich wird ein Prozentsatz von Spieler*innen angezeigt, die diese Trophäe ebenfalls erhalten haben. Diese Gamification-Elemente sollen zum spielerischen Wettstreit führen und dadurch die Motivation zusätzlich steigern.

Zur Demonstration und Evaluierung der Lernplattform wurde der Beispielkurs „Crocos Bruder“ implementiert, mit welchem grundlegende Programmierkonzepte von JavaScript für Programmier- und JavaScript-Anfänger*innen vermittelt werden. Der/Die Benutzer*in spielt dabei ein 2D-Adventure-Spiel und muss dem Protagonisten Croco bei der Rettung seines Bruders helfen, welcher zu Beginn der Geschichte entführt wird und im letzten Level des Spiels gerettet werden kann. In jedem Level wird Croco vor ein neues Problem gestellt, welches auf den vorangegangenen Leveln aufbaut und mit den erlernten Programmierkonstrukten lösbar ist. Das neu erworbene Wissen baut so auf dem vormals erlangtem auf (vgl. [Me02]) und die Relevanz dessen wird durch konkrete Anwendung und Kontextbeziehungen klar (vgl. Kellers Relevance [Ke87]). Folgende JavaScript-Themen werden in „Crocos Bruder“ behandelt: Syntax und Struktur, Kommentare, Datentypen (strings, numbers, booleans), Variablen, If und If-Else, Komparatoren, logische Operatoren, Funktionen, for-Schleifen. Die Themenwahl, deren Reihenfolge und die Lehrtexte sind an das Lehrbuch „JavaScript for Kids: A Playful Introduction to Programming“ [Mo14] angelehnt und für die Lernplattform angepasst. Die Themen decken zudem einen Teil der von Kazimoglu et al. [Ka12] und Muratet et al. [Mu09] identifizierten Programmiergrundlagen, wie konditionale Logik und Schleifen, ab.

4 Methodik und Ergebnisse der Evaluation

Um die eingangs vorgestellte Forschungsfrage zu beantworten, wurde das Lernsystem PLAY & CODE an mehreren Schulen in Berlin mit Schüler*innen unterschiedlichen Alters und mit wenig JavaScript-Erfahrungen evaluiert. Quantitative Daten wurden schriftlich durch standardisierte Online-Fragebögen und zudem automatisiert durch das System erfasst. Um den Lernerfolg der Schüler*innen feststellen zu können, wurde ein Pre-/Posttest durchgeführt. Hierbei wurden Fragen (16 Fragen im Pretest, 17 Fragen im Posttest) zu den grundlegenden Themenbereichen gestellt, welche auch im evaluierten PLAY & CODE Kurs „Crocos Bruder“ behandelt wurden (vgl. Kapitel 3.5). Im Pretest wurde zudem nach dem Alter, dem Geschlecht und der Spielhäufigkeit pro Woche bzw. Monat gefragt. Nach Abschluss des Posttests füllten die Schüler*innen einen Evaluationsfragebogen aus. Während der Benutzung von PLAY & CODE wurde außerdem die Bearbeitungsdauer und die Versuchsanzahl pro Lektion pro Benutzer*in gespeichert. Die Fragebögen (Fragen und Antwortskalen) wurden selbst erstellt, da keine getesteten Skalen hierzu vorliegen.

Im Zeitraum vom 15.02. bis zum 09.03.2018 haben sich bei PLAY & CODE 140 eindeutige Benutzer*innen registriert.[4] Die Zusammenführung von Pretest, Posttest und den Evaluationsfragebogen ergab 50 eindeutige, bereinigte Fälle. Hierbei handelt es sich um Benutzer*innen im Alter von 12 bis 18 Jahre. Um die Differenz zwischen Pre- und Posttest festzustellen, wurde zu den Antworten jedes Themengebiets in beiden Datensätzen ein gerundeter Durchschnittswert ermittelt. Hatte der/die Schüler*in die Hälfte oder mehr der

[4] Diese Zahl enthält auch Benutzer*innen, die nicht an der Befragung teilgenommen hatten und unaufgefordert oder auf Empfehlung von Anderen die Seite besuchten.

Antworten richtig beantwortet, wurde das Themengebiet als korrekt bearbeitet markiert, andernfalls als falsch. Aus den so entstandenen acht Variablen pro Datensatz wurden Summen berechnet, die über den Anteil von „richtig“ beantworteten Themenbereichen Aufschluss geben. Die Differenz stellt den Wissenszuwachs dar.

Im Evaluationsfragebogen wurden außerdem Fragen zur Usability gestellt. Die Bedienbarkeit wurde dabei mit durchschnittlich 1,9 als „gut“ bewertet. Das Design wurde von den Benutzer*innen im Mittel mit 1,4 bewertet. Insgesamt wurde PLAY & CODE mit einer 1,7 bewertet. Danach gefragt, ob die Schüler*innen gerne weitere Programmierinhalte mit Spielen lernen würden wollen, gaben 92 Prozent (46 Teilnehmende) „Ja“ an, 8 Prozent gaben „Nein“ an.

4.1 Qualität und Adaptivität der Lehrtexte

Die Lehrtexte sind bei PLAY & CODE essenziell für den Lernerfolg der Benutzer*innen. Es stellte sich daher die Frage, wie die Qualität die Lehrtexte des Evaluationskurses „Crocos Bruder“ eingeschätzt wird und wie sich diese auf den Lernerfolg der Benutzer*innen auswirkt. PLAY & CODE passt zudem die Länge der Lehrtexte an den geschätzten Wissensstand der Benutzer*innen an. Es sollte daher überprüft werden, wie die Textlänge bewertet wurde. Die Datenanalyse ergab, dass die Textlänge von 72 Prozent der Schüler*innen als „genau richtig“, von 24 Prozent als „zu lang“ und von 4 Prozent als „zu kurz“ bewertet wurde. Für den Großteil der Benutzer*innen hatten die Texte somit die richtige Länge. Das könnte einen Hinweis darauf geben, dass die Schätzung des Wissensstands der Benutzer*innen und die Adaption der Lerntexte durch das System das gewünschte Ergebnis erzielte. Für immerhin 24 Prozent waren die Texte jedoch zu lang. Für diese Gruppe sollten möglicherweise die Texte optimiert und in der Gesamtlänge gekürzt als auch eine feinstufigere Adaptivität der Texte eingeführt werden.[5] Die Textverständlichkeit wurde ebenfalls positiv, mit einer Durchschnittsschulnote von 2,1, bewertet.[6]

Bemerkenswert war zudem, dass keine signifikante Korrelation zwischen dem Alter bzw. der Vorerfahrung und dem Textverständnis, sowie zwischen dem Alter bzw. der Vorerfahrung und der Gesamtsumme an gewonnenen Trophäen festgestellt werden konnte. Die Texte scheinen sich daher für alle Benutzer*innen gleichermaßen zu eignen, was an der Strukturierung der Texte sowie deren Adaptierung an den Wissensstand der Benutzer*innen liegen könnte. Somit wird eine gleichbleibende Textverständlichkeit bei sinkendem Erfahrungswert durch einen steigenden Detailierungsgrad und einer steigenden Länge des Textes erreicht.

[5] Mit den hier erfassten Variablen lässt sich diese Gruppe nicht eindeutig klassifizieren. Um eine bessere Adaptivität anbieten zu können, müssten daher zukünftig die Eigenschaften dieser Benutzer*innen genauer erfasst werden.

[6] 20 Prozent der Benutzer*innen haben die Textverständlichkeit mit der Schulnote „sehr gut“, 62 Prozent mit der Schulnote „gut“ bewertet. 10 Prozent empfanden die Textverständlichkeit als „befriedigend“. Jeweils 4 Prozent gaben die Note „ausreichend“ und „mangelhaft“.

4.2 Motivation, Kreativität und Spaß

Motivation und Spaß sind wichtige Faktoren für den Lernerfolg. Auch die Möglichkeit, Probleme kreativ lösen zu können, ist eine Bedingung für erfolgreiches Lernen. In PLAY & CODE sollen diese Faktoren vor allem durch den Einsatz von Serious Games und Trophäen als Gamification-Element erzeugt werden. Der Erfolg dieser Komponenten wurde daher überprüft. Außerdem wurde untersucht, wie schwierig die Schüler*innen die Aufgaben empfanden und wie sie die Relevanz des Gelernten bewerten.

Nach dem Spaß gefragt, gaben 36 Prozent der Schüler*innen an, dass sie „sehr viel Spaß" hatten, weitere 54 Prozent hatten „viel Spaß". Nur 8 Prozent der Benutzer*innen gaben an, dass sie „wenig Spaß" hatten. Die Antwortvorgabe „überhaupt kein Spaß" wurde nicht ausgewählt. Im Durchschnitt wurde der Spaß mit 1,8 bewertet (Skala von 1 – „sehr viel Spaß" bis 4 – „überhaupt kein Spaß").[7] Bemerkenswert ist dabei die Korrelation zwischen der Dauer der Bearbeitung und dem Spaß (Einseitig, Kendalls τ, $p < 0{,}05$, Korrelation: -0,295).[8] Anders als zunächst angenommen, nahm der Spaß mit steigender Dauer nicht ab, sondern zu. Dies könnte einen Hinweis darauf geben, dass die Schüler*innen daran interessiert waren, die Level, trotz der damit verbundenen Herausforderung, zu lösen. Vermutlich sind diese Effekte aber bis zu einer gewissen Dauer begrenzt und eine Frustration stellt sich danach dennoch ein.

Gefragt nach der Motivation durch Trophäen, gaben 26 Prozent der Schüler*innen „sehr motivierend" und 46 Prozent der Schüler*innen „motivierend" an. 22 Prozent der Schüler*innen fanden die Trophäen „wenig motivierend", 6 Prozent „überhaupt nicht motivierend". Im Durchschnitt empfanden die Schüler*innen die Trophäen als „motivierend" (2,3 bei Bewertungsskala 1 bis 4) mit einer Standardabweichung von 1,2. Die lediglich „gute" Bewertung verwundert insofern, als dass 90 Prozent aller Teilnehmenden mehrmals pro Woche oder häufiger Videospiele spielen und daher angenommen wurde, das Gamification-Elemente, wie sie auch in Videospielen vorkommen, mit einer deutlich positiveren Rückmeldung der Benutzer*innen verbunden ist. Möglicherweise war der unmittelbare Nutzen der Trophäen nicht klar und deutlich ersichtlich. Dieser Zusammenhang sollte zukünftig weiter untersucht werden.

Mit dem Spaßempfinden korreliert auch der Wissenszuwachs. Schüler*innen, die selbst einschätzen, viel dazugelernt zu haben, hatten nach eigenen Aussagen auch mehr Spaß beim Spiel (Skala 1 – "sehr viel Spaß" bis 5 „überhaupt kein Spaß", Einseitig, Pearson, $p < 0{,}05$, Korrelation: -0,38). Dies gibt einen Hinweis darauf, dass sich die von Keller [Ke87] beschriebene Satisfaction eingestellt hat: Durch sichtbare Erfolge wurde mehr Spaß empfunden.

Um die Kreativität der Benutzer*innen bei der Lösung der Aufgaben zu ermitteln, wurden Levenshtein-Distanzen zwischen der Abgabe der Benutzer*innen und der Musterlösung

[7] Gestellt wurde die Frage „Wieviel Spaß hat Dir das Lernen mit Spielen gemacht?" mit folgenden Antwortmöglichkeiten „sehr viel Spaß", „viel Spaß", „wenig Spaß", „überhaupt kein Spaß".

[8] Da fünf Schüler*innen ein leeres Namenskürzel im System angegeben haben, ist hier N = 45.

für die jeweilige Aufgabe errechnet und summiert (N = 45). Je höher die Gesamtsumme, desto kreativer war der/die Benutzer*in bei der Lösung der Aufgaben. Ein Zusammenhang zwischen der Kreativität der Benutzer*innen und dem Spaß konnte zwar nicht festgestellt werden, jedoch hängen Kreativität und die jeweilige Ergebnissumme zusammen: Je kreativer die Schüler*innen bei der Lösung ihrer Aufgaben waren, desto bessere Ergebnisse konnten sie erzielen (Pearson, $p < 0{,}01$, Korrelation: 0,42). Insbesondere bei Aufgaben mit optionalen Zielen kreierten viele Schüler*innen individuelle Lösungen, welche ebenfalls zum bestmöglichen Ergebnis führten. Der Schwierigkeitsgrad der Aufgaben wurde von 80 Prozent der Schüler*innen als „genau richtig" bewertet. 8 Prozent fanden die Aufgaben „zu leicht", 12 Prozent „zu schwer". Die Aufgaben waren im Sinne Kellers [Ke87] „Confidence" demnach fordernd, aber machbar.

4.3 Lernerfolg

Nachdem im vorherigen Abschnitt die Faktoren untersucht wurden, die einen Einfluss auf den Lernerfolg haben, soll nun der Lernerfolg der Schüler*innen selbst untersucht werden. Hierzu werden die Ergebnisse des Posttest-Quiz näher betrachtet und die Frage gestellt, ob die Schüler*innen das Erlernte im Posttest-Quiz anwenden konnten.

Der Wissenszuwachs wurde als die Differenz zwischen dem Pretest-Ergebnis und dem Posttest-Ergebnis berechnet. Die Ergebnisse der Tests konnten jeweils Werte zwischen 0 und 8 annehmen. Wurden 8 Punkte erreicht, konnten jeweils mehr als die Hälfte der Fragen der acht Themenbereichen korrekt beantwortet werden. Bei 0 Punkten wurde kein Themenbereich korrekt beantwortet. Angestrebt wurde eine möglichst große Differenz als Wissenszuwachs zwischen dem Pretest- und dem Posttest-Ergebnis.

Im Durchschnitt konnten die Schüler*innen im Posttest 4,2 Fragenkategorien mehr als im Pretest korrekt beantworten und damit ihr Ergebnis des Pretests um 50 Prozent verbessern.[9] Dabei wurden insbesondere Fragen zu Kommentaren, Vergleichsoperatoren, Datentypen und If-Abfragen richtig beantwortet.[10] Die Schüler*innen haben somit eine grundsätzliche Vorstellung von Programmierung und Wissen um die Datentypen und konditionalen Verzweigungen in JavaScript durch das Spiel erlangt.

[9] Die Evaluierung fand pro Klasse in einer Schuldoppelstunde (90 Minuten) statt. Die Schüler*innen konnten in dieser Zeit durchschnittlich 10 von 12 Lektionen mit einer Standardabweichung von 2,043 absolvieren (N = 45). Da die 12. Lektion eine zusammenfassende Aufgabe darstellt, haben die Schüler*innen im Durchschnitt zehn der elf Themenbereiche bearbeitet, lediglich for-Schleifen (Lektion 11) wurden von 60 Prozent der Teilnehmenden nicht gelernt.

[10] Kommentare: 40 korrekte und 10 falsche Antworten; Vergleichsoperatoren: 40 korrekte und 10 falsche Antworten; Datentypen: 43 korrekte und 7 falsche Antworten; If-Abfragen: 40 korrekte und 10 falsche Antworten; Variablen: 24 korrekte und 26 falsche Antworten; for-Schleifen: 16 korrekte und 34 falsche Antworten; Funktionen: 21 korrekte und 29 falsche Antworten; Logik-Operatoren: 11 korrekte und 39 falsche Antworten.

5 Zusammenfassung und Fazit

Die gesellschaftliche Bedeutung der Informatik nimmt kontinuierlich zu, gleichzeitig fehlt es an Fachkräften. Vielen erscheint Programmierung als zu abstrakt und mühsam. Es müssen deshalb neue Wege gefunden werden, Programmieranfänger*innen zu motivieren und das notwendige Fachwissen praxisnäher zu vermitteln. Es stellte sich daher die Frage, ob mithilfe eines Serious Games basierten Lernsystems Motivation und Lernerfolg für grundlegende Programmierkenntnisse gesteigert werden können. Zur Beantwortung dieser Frage wurde die Serious Games basierte Lernplattform PLAY & CODE entwickelt.

Um die Wirksamkeit dieser Lernplattform zu überprüfen, wurde eine Evaluation an mehreren Schulen in Berlin durchgeführt. Den Schüler*innen wurden mit der Lernplattform Grundlagen von JavaScript vermittelt. Dabei wurde gezeigt, dass die Schüler*innen bei der Lösung der Aufgaben motiviert waren und Spaß hatten. Mittels eines Pre- und Posttests wurde festgestellt, dass die Schüler*innen nach dem Kurs 50 Prozent mehr Fragen zu den gelehrten Themenbereichen korrekt beantworten konnten als davor. Die Schüler*innen hatten nach dem Spiel eine genauere Vorstellung von Programmierung und verfügten über einige Grundlagen von JavaScript. Die Forschungsfrage kann daher positiv beantwortet werden: Serious Games basierte Lernsysteme wie PLAY & CODE sind ein wirkungsvolles Mittel zur Vermittlung von Programmiergrundlagen.

Um die Qualität der Plattform noch zu steigern, wird in Zukunft eine feinstufigere Adaptivität der Texte in Erwägung gezogen. Untersucht werden sollte auch, wie die Motivation durch Trophäen weiter gesteigert werden kann. Auch bei den Erhebungsinstrumenten gibt es Optimierungsbedarf. Zukünftig sollten in den Fragebögen die Namensfelder erzwungen werden, um eine höhere Anzahl an vollständigen Datensätzen zu erzielen. Außerdem werden in kommenden Evaluierungen im Pre- und Posttest jeweils die gleiche Anzahl an Fragen pro Themenbereich gestellt. Damit kann der Wissenszuwachs genauer bestimmt werden. Zudem wurde die Lernplattform ausschließlich in Informatik-Schulklassen evaluiert. Da angenommen werden kann, dass diese Schüler*innen ein gesteigertes Verständnis für Programmierung haben, wurde das Ergebnis dadurch womöglich verzerrt. PLAY & CODE sollte daher zukünftig auch von Personen aus dem Hobby- und Selbstlernbereich evaluiert werden.

6 Danksagung

Ich danke Herrn Prof. Dr. Pinkwart für seine wertvollen Kommentare und Anmerkungen bei der Erstellung dieses Beitrags.

Literaturverzeichnis

[Ch15] Cheng, M. et. al.: The use of serious games in science education: a review of selected empirical research from 2002 to 2013. Journal of Computers in Education, 02/15: S. 353-375, 2015.

[Dö16] Dörner, R.; Göbel, S.; Effelsberg, W.; Wiemeyer, J.: Serious Games. Foundations, Concepts and Practice, Springer International Publishing, Basel, 2016.

[GEM13] Girard, C.; Ecalle, J.; Magnan, A.: Serious games as new educational tools: how effective are they? A meta-analysis of recent studies. Journal of Computer Assisted Learning, 29/13: S. 207-219, 2012.

[He17] Heublein, U. et. al.: Motive und Ursachen des Studienabbruchs an baden-württembergischen Hochschulen und beruflicher Verbleib der Studienabbrecherinnen und Studienabbrecher. DZHW Projektbericht, 6/17: 2017.

[Hu07] Hubwieser, P.: Didaktik der Informatik. Grundlagen, Konzepte, Beispiele, 3. Auflage, Springer-Verlag Berlin Heidelberg, Berlin, 2007.

[IW17] IW, Institut der deutschen Wirtschaft, https://www.iwkoeln.de/presse/iw-nachrichten/beitrag/oliver-koppel-mint-fachkraefte-verzweifelt-gesucht-350061.html, Stand: 09.03.2019

[Ka12] Kazimoglu, C. et. al.: A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming. Procedia - Social and Behavioral Sciences, 47/12: S. 1991-1999, 2012.

[Ke87] Keller, J. M.: Development and use of the ARCS model of instructional design. Journal of instructional development, 3/87, S. 2-10, 1987

[Me02] Merrill, M. D.: First principles of instruction, Educational technology research and development, 3/02: S. 43-59, 2002.

[Mo14] Morgan, N.: JavaScript for Kids. A Playful Introduction to Programming, No Starch Press, San Francisco, 2014.

[Mu09] Muratet, M. et. al.: Towards a serious game to help students learn computer programming. International Journal of Computer Games Technology, 3/09: S. 1-12, 2009.

[Sa06] Savery, J. R.: Overview of problem-based learning: Definitions and distinctions. Interdisciplinary Journal of Problem-Based Learning, 1/06: S. 9-20, 2006.

[Sc13] Schäfer, A. et. al.: M.: From boring to scoring. a collaborative serious game for learning and practicing mathematical logic for computer science education. Computer Science Education, 23/13: S. 87-111, 2013.

[TI19] TIOBE Programming Community Index, https://www.tiobe.com/tiobe-index/, Stand: 09.03.2019

[VDC11] Vandewaetere, M.; Desmet, P.; Clarebout, G.: The contribution of learner characteristics in the development of computer-based adaptive learning environments. Computers in Human Behavior, 27/11: S. 118-130, 2011.