

Einsatz einer Online-Programmierplattform in der Präsenzlehre – ein Erfahrungsbericht

Torge Hinrichs¹ und Axel Schmolitzky²

Abstract: Online-Programmierplattformen bieten umfangreiche Möglichkeiten, die individuellen Programmierfähigkeiten zu verbessern. Durch die Verwendung einer solchen Plattform in einer Präsenzveranstaltung konnte die Motivation der Teilnehmer stark erhöht werden. Es wurden aber auch etliche Defizite deutlich, die in zukünftigen Plattformen, die für die Hochschullehre zugeschnitten sind, adressiert werden sollten.

Keywords: Programmierwettbewerb, automatische Bewertung von Aufgaben, Softwaretechnik, Code-Qualität

1 Einleitung

In den Bachelor-Studiengängen der Informatik an der HAW Hamburg hat die Programmierlehre seit jeher einen besonderen Stellenwert. Trotz dieser Betonung im gesamten Curriculum und obwohl nur wenige Module ohne Aufgaben auskommen, die Programmierkenntnisse erfordern, fällt es vielen Studierenden auch in höheren Semestern noch schwer, Lösungen für komplexe oder abstrakte Problemstellungen geeignet in Quellcode zu modellieren. Im fünften Semester sehen alle Bachelor-Studiengänge der Informatik ein unbenotetes *Projekt* vor, in dem die Studierenden weitgehend eigenständig eine umfangreichere Problemstellung bearbeiten sollen, idealerweise im Team. Den fachlichen Kontext bestimmen die Lehrenden der Informatik, indem sie mehrere Projekte anbieten, die von den Studierenden gewählt werden können.

In einem dieser Projekte haben wir die Teilnehmer angeleitet, an verschiedenen Trainingsaufgaben und Wettbewerben einer Online-Programmierplattform teilzunehmen. Neben der individuellen Verbesserung der Programmierfähigkeiten sowie der Fähigkeit, abstrakte Problemstellungen mit Software zu lösen, war ein weiteres Ziel, die Teilnehmer wiederholt zu fordern, die von ihnen gewählten Lösungsansätze einander seminaristisch vorzustellen. Durch diese regelmäßigen Präsenztermine sollte zusätzlich die Fähigkeit trainiert werden, über Entwürfe mit Kommilitonen zu diskutieren und von guten Designs anderer zu profitieren. Ein Kerngedanke war somit, dass komplexe Sachverhalte spielerisch thematisiert und als weniger schwerfällig wahrgenommen werden.

In diesem Erfahrungsbericht stellen wir unser Konzept für den Einsatz einer Online-Programmierplattform in einer Lehrveranstaltung sowie erste Erfahrungen damit vor.

¹ HAW Hamburg, Dep. Informatik, Berliner Tor 7, 20099 Hamburg, torge.hinrichs@haw-hamburg.de

² HAW Hamburg, Dep. Informatik, Berliner Tor 7, 20099 Hamburg, axel.schmolitzky@haw-hamburg.de

2 CodinGame: eine Online-Programmierplattform

Da Programmierfähigkeiten durch die zunehmende Digitalisierung unserer Gesellschaft eine Schlüsselkompetenz für immer mehr Stellenbeschreibungen darstellen, erfreuen sich *Online-Programmierplattformen* immer größerer Beliebtheit. Die Nutzer dieser Plattformen können zwischen verschiedenen Modi wählen, um ihre Programmierfähigkeiten auf unterschiedlichen Niveaus zu trainieren. Dazu gibt es eine Fülle an Übungsaufgaben, die einfache Schleifen bis hin zu anspruchsvollen Algorithmen erfordern. Die Aufgaben werden in der Regel in einem beschreibenden Text formuliert und üblicherweise mit einem oder mehreren Beispielen genauer erläutert.

Für die Lösung ist die Wahl der Programmiersprache unerheblich, da die Eingaben der notwendigen Größen, wie zum Beispiel der aktuelle Zustand der Umgebung in einer Simulation, über die *Standard-Eingabe* eingelesen und die Ausgaben der Lösungen auf der *Standard-Ausgabe* ausgegeben werden müssen (*Standard-IO*). Die Plattform vergleicht die Ausgaben mit erwarteten Ergebnissen. Die Aufgaben werden häufig als Minispiele aufgebaut, die rundenbasiert durchlaufen werden. Der Spielablauf wird dabei in einer *zweidimensionalen Darstellung* der Spielwelt visualisiert.

CodinGame³ verfügt zusätzlich über verschiedene Wettkampfmodi, in denen Nutzer direkt gegeneinander antreten können. Im *Arena-Modus* programmiert der Nutzer das Verhalten eines Bots, der dann in einer Arena gegen die Implementierungen von anderen Nutzern antritt. Aufgrund der rundenbasierten Spielweise mit Visualisierung können Nutzer jede Runde eines jeden Spiels genau nachvollziehen. Die Nutzer werden für jedes Spiel in Ligen mit aufsteigendem Anspruch einsortiert. Dabei werden die Aufgaben teils erweitert und es können neue Funktionalitäten hinzukommen, die neue und komplexere Strategien ermöglichen bzw. erfordern.

CodinGame bietet Einzelpersonen umfangreiche und individuelle Unterstützung bei der Verbesserung ihrer Programmierfähigkeiten. Die eingesetzten Wettkampfkonzpte fördern für viele Personen den Spaß am Programmieren, die spielerischen Inhalte wirken durch die interaktive Visualisierung eines Spielverlaufs in Duell-Form stark motivierend.

Aus softwaretechnischer Sicht bestehen aber durchaus Defizite:

- Der „EVA-Flaschenhals“ (Standard-Eingabe, Verarbeitung, Standard-Ausgabe) bietet den Vorteil der Sprachunabhängigkeit, bewirkt aber einen Fokus auf algorithmisch geprägte Aufgaben; die Architektur der implementierten Lösungen ist irrelevant.
- Die Code-Qualität wird vernachlässigt; das Einhalten von Quellcode-Konventionen oder von Regeln des guten Entwurfs spielt keine Rolle.
- Die Plattformen ist auf Einzelpersonen ausgelegt: durch die integrierte Online-IDE, die die Eingabe in nur einem Textfenster gestattet, wird es erschwert, Lösungen im Team zu erarbeiten, zu modularisieren und zusammenzuführen.

³<https://www.codingame.com/>, letzter Abruf Juni 2019

Diese Defizite schmälern keinesfalls den Erfolg der Plattform, sind jedoch relevant für den Einsatz in einer Projekt-Lehrveranstaltung eines Informatik-Studiums.

3 Einsatz in einem Lehre-Projekt

Für die Pilot-Projektveranstaltung, in der wir den Einsatz einer Online-Programmierplattform testen wollten, hatten wir uns als Lernziel neben der Verbesserung der Programmierfähigkeiten gesetzt, dass die Studierenden besser darin werden sollen, sich über Softwareentwürfe gewinnbringend auszutauschen. Für das letztere Ziel schien uns CodinGame aufgrund der Visualisierung des Spielverlaufs besonders geeignet, weil diese den Studierenden die Analyse und das Verständnis unterschiedlicher Lösungen erleichtert. Der Umfang des Moduls *Projekt* im fünften Semester unserer Bachelor-Studiengänge beträgt 9 Credit Points, den studentischen Aufwand kalkulieren wir deshalb mit etwa 12 Stunden pro Woche. Für die Dozenten werden lediglich 3 LVS angerechnet, so dass die Kontaktzeit nur einen kleinen Anteil des studentischen Gesamtaufwandes ausmacht.

3.1 Projektablauf

Unser Projekt begann mit einer zweiwöchigen Einarbeitungsphase in CodinGame. Dazu haben die Studierenden in Einzelarbeit von uns ausgewählte Einstiegsaufgaben mit aufsteigender Komplexität gelöst, um sich mit den Besonderheiten der Plattform vertraut zu machen. Im nächsten Schritt wurde die Einzel- durch Teamarbeit ersetzt, indem Teams mit jeweils vier Personen gebildet wurden. Die Teamkomposition erfolgte zufällig; es gab lediglich die Maßgabe, dass in jedem Team verschiedene Studiengänge vertreten sind, weil die Programmierlehre in den ersten Semestern teilweise sehr unterschiedlich ist.

Die Teams mussten an einem von uns vorgegebenen Arenaspiel teilnehmen, mit dem Ziel, möglichst weit in den Ligen aufzusteigen. Da Teamarbeit durch die Plattform nicht vorgesehen ist, mussten sich die Teams selbst organisieren. Damit wir einen Überblick über den Fortschritt erhielten, haben wir wöchentliche StandUp-Meetings mit den einzelnen Gruppen durchgeführt, in denen diese den aktuellen Stand der Bearbeitung sowie den Anteil der jeweiligen Mitglieder am Fortschritt vorstellten. Den „Erfolg“ eines Teams konnten wir zusätzlich durch seinen automatisch ermittelten Liga-Rang bemessen; eine höhere Platzierung korrelierte systembedingt stark mit einer besseren Lösung.

Nach zwei Wochen wurde ein Plenumstermin durchgeführt, in dem die Gruppen ihre Strategien und verwendeten Algorithmen und Datenstrukturen in Vorträgen (30 Minuten) vorstellten. Anschließend erhielten die Gruppen erneut zwei Wochen Zeit (mit einem StandUp in der Mitte), um die beim Austausch gelieferten Anregungen in die eigene Lösung zu integrieren und idealerweise in die höchste Liga aufzusteigen. Abschließend gab es erneut einen Plenumstermin, diesmal zum Austausch in 15-minütigen Vorträgen über die jeweiligen Fortschritte.

Dieser Prozess wurde im Lauf des Semesters dreimal mit verschiedenen Arenaspielen durchlaufen. Des Weiteren haben die Studierenden nach Ende der Bearbeitung eines Spiels ihren Quellcode an eine andere Gruppe zum Code-Review weitergeben. Die Ergebnisse wurden auch im Plenum diskutiert.

3.2 Beobachtungen

Die Visualisierung des Spielverlaufs (bzw. der Aufgabenstellung bei den Übungsaufgaben) von CodinGame ermöglichte den Studierenden einen leichten Einstieg in die Plattform. Die implementierte Lösung kann im Detail untersucht und verstanden werden. Die automatische Prüfung der Aufgabe konnte dazu genutzt werden, direktes Feedback zu erhalten. Nach Bearbeitung einer Aufgabe haben sich die Studierenden häufig die Lösungen anderer Nutzer angesehen und diskutiert.

Über den Verlauf des Projekts konnten wir eine Änderung der Denkweise bzw. der Bearbeitungsweise der Studierenden beobachten. So arbeiteten sie beim ersten Spiel überwiegend „drauf los“ und hatten Probleme sich im Team zu koordinieren. Unübersichtlicher Quellcode und ein langsamer Aufstieg in den Ligen waren die Folge. Des Weiteren wurden Probleme in der Abstraktion der Problemstellung deutlich. Der erste Plenumstermin zwang die Studierenden, ihre Ansätze zu ordnen und in einen Vortrag zu bringen. Bei den folgenden Arenaaufgaben war ein deutlich strukturierterer Ansatz mit überlegten Architekturen erkennbar. Die Teams waren in der Lage, die Aufgabenstellung in Basisbestandteile zu zerlegen und Diskussionen über das Verhalten auf einer abstrakteren Ebene zu führen. Außerdem waren der Austausch bzw. das Review der Quellcodes anderer Gruppen nach Aussage der Studierenden besonders hilfreich. Das Konfrontieren mit anderen Denkweisen brachte die Studierenden dazu, stärker über ihre eigene Lösung zu reflektieren.

4 Fazit und Ausblick

Der Einsatz einer Online-Programmierplattform in einer Präsenzveranstaltung war aus unserer Sicht ein Erfolg. Die von uns verwendete Plattform CodinGame bietet eine attraktive Visualisierung der Lösungen und einen Arena-Modus, dessen „Programmieren gegen Andere“ stark motivierend wirkte. Der Wettbewerb und insbesondere das Präsentieren der eigenen Lösungsansätze brachten die Studierenden dazu, strukturierter über ihre Lösungen zu reflektieren. Ergänzt durch das systematische Peer-Feedback über den produzierten Quellcode konnte eine effektive Lernerfahrung geboten werden. Die automatische Überprüfung der Lösungen war ebenfalls für uns als Dozenten hilfreich, weil weniger Zeit für händische Prüfungen anfiel und der Schwerpunkt der Diskussion auf die Kern-Abstraktionen der Lösungen gelegt werden konnte. Das entwickelte Konzept wird als Blaupause für weitere Projekte dieser Art dienen.