

Learning by Questing

Integration von gamifizierten Quests zur Steigerung der Lernmotivation in der Hochschulbildung

Verfassende (André Matutat; Malte Reinsch; André Kirsch; Birgit Christina George; Carsten Gips; Hochschule Bielefeld)

- **Kurzfassung (ca. 100-150 Wörter)**

Gamification kann ein valides Mittel zur Steigerung der Motivation von Lernenden im Hochschulbereich sein. Dieses Paper stellt ein Tool vor, das Lehrenden ermöglicht, Übungsaufgaben als Quests in ein 2D-Rollenspiel zu integrieren, ohne über tiefgreifendes Wissen in der Software- oder Videospieldentwicklung zu verfügen. Hierfür wurde eine Beschreibungssprache definiert, in der Aufgaben definiert werden können. Diese werden automatisch in verschiedene Spielszenarien umgewandelt, die von den Lernenden innerhalb des Spiels gelöst werden müssen. Darüber hinaus wird ein Ansatz beschrieben, bei dem Petri-Netze zur Definition von Aufgabenbeziehungen und alternativen Aufgabensequenzen genutzt werden, um den Ansatz des selbstgesteuerten Lernens zu verfolgen.

- **Keywords (3-5 Keywords)**

Gamification, Self-Paced Learning, Petri-Netze, DSL

1 Kapitelüberschriften

Wir haben im Studiengang Informatik Module mit hohem Programmieranteil erfolgreich auf ein selbstgesteuertes Lernkonzept mit Gamification-Ansatz umgestellt. Die Studierenden erstellen in den Übungsaufgaben Code, um schrittweise in einem vorgegebenen Java-Framework ein eigenes 2D-Rollenspiel mit individuellen Spielelementen zu entwickeln. Laut Schreuders et al. [1] erhöht der Spielspaß die Motivation der Studierenden bei der Bearbeitung der Programmieraufgaben deutlich. Gleichzeitig wird der starken Heterogenität in Bezug auf Vorwissen Rechnung getragen (vgl. [2]). Dieses Paper erweitert den Game-based Learning-Ansatz und zeigt, wie Lehrende selbst Aufgaben für ein 2D-Rollenspiel formulieren können. Dabei werden fachliche Aufgaben als Quests im Spiel repräsentiert. Studierende anderer Fachrichtungen müssen keinen Quellcode schreiben, sondern lösen die vorgegebenen Aufgaben durch Spielen.

Ähnliche Projekte mit dem Ziel, Spiele zum Lernen im (Hoch-)schulkosmos einzusetzen, fokussieren sich auf modul- oder fachbereichspezifische Aufgabentypen [3]–[5]. Dabei konnte eine erhöhte Motivation der Lernenden durch das spielerische Lernen festgestellt werden; dieses Projekt geht darüber hinaus und setzt eine fachübergreifende Toolchain zur Gamifizierung von Übungsaufgaben um. Dafür wird das Framework so erweitert, dass mithilfe einer Beschreibungssprache Aufgaben in ein Spiel integriert werden können.

Die Lösungen und Lösungswege im Spiel werden über die im Framework implementierten Mechanismen bewertet. Analog zu den Ideen in Schreuders et al. [1] kann ein Akteur im Spiel diverse Belohnungen durch Lösen verschiedener Aufgaben(-typen) sammeln.

2 Vergleichbare Arbeiten

In der Lehre wurden bereits einige Erfahrungen zum Thema Gamification gesammelt. So haben Coller et al. [3] ein Rennspiel für das Modul Maschinenbau entwickelt, bei dem mit manuellem Schalten Bestzeiten gefahren werden mussten. Um möglichst gute Zeiten zu erreichen, musste der optimale Schaltpunkt getroffen werden. Um den optimalen Schaltpunkt zu finden, mussten Studierende die in der Vorlesung gelehrt Techniken anwenden. Die Ergebnisse deuten darauf hin, dass die Studierenden eine höhere intellektuelle Intensität, intrinsische Motivation und insgesamt mehr Engagement erfahren, wenn sie mit dem Rennspiel im Vergleich zu traditionellen Methoden für Hausaufgaben und Unterricht im Bereich Maschinenbau arbeiten.

Zum Lehren von UML-Modellierung wurde von Jurgelaitis et al. [4] ein Moodle-Kurs entwickelt, in dem Studierenden aufgeteilt in mehrere freischaltbare Level die Modellierung verschiedener UML-Diagrammtypen lernen. Weitere Features des Kurses sind Rewards, Leaderboards und Trading. Die Evaluation hat gezeigt, dass die Motivation der Studierenden stark gesteigert werden konnte. Eine geplante Erweiterung ist die automatische Validierung der eingereichten UML-Diagramme. In unserem Fall können die Aufgaben so designt werden, dass sie automatisch bewertet werden. Außerdem können eigene Bewertungsfunktionen implementiert werden.

Zwar nutzen beide genannten Arbeiten Gamification, um Lerninhalte eines konkreten Themengebiets aufzubereiten. Die erzeugten Artefakte eignen sich ohne größere Anpassungen jedoch nicht für einen Themengebiets-übergreifenden Einsatz. Das in diesem Paper vorgestellte Projekt ermöglicht das Erstellen von Übungsaufgaben aus verschiedenen Themengebieten.

Für Sustainable Entrepreneurship Education wurde von Frenz et al. [5] ein Escape Room entwickelt. Der Escape Room wurde als E-Book umgesetzt, in dem Texte, Bilder und Videos integriert sind. Quizze werden als Online-Tools über Links eingebunden, über die Studierende Codes zum Freischalten weiterer Inhalte erhalten. Die Autoren haben erkannt, dass ein E-Book Grenzen aufweist, da es nicht für diesen Einsatz konzipiert wurde. So kann zum Beispiel das Überspringen von Aufgaben nicht verhindert werden. Unsere Anwendung nutzt daher zur Kontrolle des Lernflusses ein Petri-Netz, mit dem der Aufgabenverlauf genau gesteuert werden kann.

3 Definition von Aufgaben mit einer Domain Specific Language (DSL)

Bei der Umsetzung einer Aufgabe als Quest ergeben sich verschiedene Herausforderungen. Zum einen sollen die fachbezogenen Inhalte im Vordergrund stehen und nicht die Spielmechaniken. Zum anderen soll das Lösen der Aufgabe Spaß machen und dadurch die Lernenden motivieren.

Außerdem sollen auch Lehrende ohne Kenntnisse von Software- oder Videospieleentwicklung auf einfache Weise eine Aufgabe formulieren können, die automatisch in das Spiel integriert wird.

Mithilfe der eigenen Beschreibungssprache (DSL) und einer Toolchain zur Umwandlung von Aufgaben in Quests sollen Lehrende bei der Integration ihrer Aufgaben unterstützt werden. Lehrende formulieren Aufgaben in der DSL, die dann in das Spiel geladen werden. Für eine minimale Aufgabendefinition muss nur der Aufgabentyp ausgewählt werden und die Aufgabenstellung sowie Lösung angegeben werden. Über die DSL-Toolchain wird die formulierte Aufgabe in eine konkrete Quest übersetzt. Angeregt durch das Konzept von Lemoine et al. [6] kann dabei eine Aufgabe automatisiert auf mehrere vordefinierte Szenarien (konkrete Spielsituationen) abgebildet werden.

Die Definitionen der Szenarien sind abhängig vom Aufgabentyp und werden ebenfalls in der DSL formuliert. Einige Szenario-Definitionen werden in einer Standardbibliothek zusammen mit der DSL ausgeliefert. Dazu gehören auch einige vordefinierte Bewertungsfunktionen zur Auswertung der Lösungen, die die Eingaben der Lernenden analysieren und bewerten. Es ist auch möglich eigene Bewertungsfunktionen in der DSL zu definieren.

```
· replacement_task chemie {
·   · description: "Führen Sie Fotosynthese durch!"
·   · initial_element_set: {10x"C", 10x"O2", 10x"H2O"}
·   · elements: {
·     · n1: ("C", "O2"),
·     · n2: {"CO2"},
·     · n3: {6x"CO2", 6x"H2O"},
·     · n4: {"C6H12O6", "6O2"}
·   · },
·   · rules: {
·     · r1: n1 -> n2,
·     · r2: n3 -> n4
·   · },
·   · answer_configuration: ["C6H12O6", "6O2"],
· }
}
```

Abbildung 1: Beispiel für eine Ersetzen-Aufgabe zum Thema Fotosynthese in der DSL. In "initial_element_set" werden die initial verfügbaren Elemente definiert. Über die in "rules" definierten Regeln können die Elementkombinationen aus "elements" zu neuen Elementen kombiniert werden. Eine Aufgabe ist korrekt gelöst, wenn die Antwort mit dem Ergebnis in "answer_configuration" übereinstimmt.

Bei der Auswahl der umzusetzenden Aufgabentypen wurde darauf geachtet, dass die Aufgabentypen unterschiedliche Aufgaben in verschiedenen Fachbereichen ermöglichen. Die Aufgabentypen sind Single-Choice, Multiple-Choice, Lückentext, Ersetzen und Zuordnen. Bei Ersetzen muss aus einer Gesamtmenge aus l Elementen eine Menge mit m Elementen durch eine zweite Menge mit n Elementen ersetzt werden. Bei Zuordnen müssen Elemente aus einer Menge A den Elementen aus einer anderen Menge B zugeordnet werden. Fachspezifische Aufgabentypen wurden nicht umgesetzt. Für die verschiedenen Aufgabentypen werden verschiedene Szenarien umgesetzt, um die Aufgaben abwechslungsreich im Spiel zu integrieren, beispielsweise das

Aufsammeln von bestimmten im Spiel verstreuten Items oder durch das Ablegen bestimmter Gegenstände in eine Kiste.

In diesem Fallbeispiel wird eine Aufgabe aus dem Bereich Chemie formuliert und in das Spiel integriert (Siehe Abb. 1). Die Lernenden müssen dabei das Reaktionsschema von Fotosynthese anwenden, indem sie schrittweise die definierten Regeln anwenden. Jede Aufgabe besitzt einen Namen, damit sie an anderen Stellen der DSL, etwa in anderen oder in verschachtelten Aufgabendefinitionen referenziert werden kann. Neben der Angabe der korrekten Antwort über „answer_configuration“ können auch die Lösungsschritte definiert werden über „answer_sequence“.

Diese Aufgabe wird im Spiel als Crafting-Szenario (Siehe Abb. 2) repräsentiert. Dabei werden die verschiedenen Elemente der Ausgangsmenge als Items zum Aufsammeln im Spiellevel verteilt. Diese Items können die Lernenden an einem “Zauberkessel” miteinander kombinieren und erhalten entsprechend den formulierten Regeln neue Items. Im Fallbeispiel würden die Items die chemischen Verkettungen aus der Aufgabenstellung repräsentieren. Ziel der Lernenden ist es, schrittweise die richtigen Items im Zauberkessel zu kombinieren, bis die Aufgabenstellung gelöst ist. Um die Aufgabe etwas schwerer zu gestalten, kann man zusätzliche Items und Regeln formulieren, die nicht für die korrekte Lösung genutzt werden sollen.



Abbildung 2: Umsetzung des Fallbeispiels im Dungeon. Studierende müssen die Zutaten in der richtigen Reihenfolge per Drag and Drop aus dem Inventar auf den Zauberkessel ziehen.

4 Steuerung des Aufgabenverlaufs mit Petri-Netzen

Zur Modellierung von Beziehungen zwischen (Teil-)Aufgaben und zur Steuerung der Reihenfolge der Bearbeitung von Aufgaben wird ein auf Petri-Netzen basierender Modellierungsansatz untersucht. Darüber können Aufgabensequenzen und alternative Pfade durch ein Level definiert werden und es können Aufgaben(-teile) als optional markiert werden. Studierende können so eigenständig die nächsten Aufgaben zur Bearbeitung auswählen.

In der DSL können Lehrende Beziehungen zwischen mehreren (Teil-)Aufgaben definieren:

- Reihenfolge: Es können (Teil-)Aufgaben definiert werden, welche zuerst bearbeitet werden müssen, um andere Aufgaben(-teile) zur Bearbeitung freizuschalten. Die Folgeaufgaben können auch bedingt durch die Qualität der Lösung der Voraufgaben definiert werden.
- Dynamik: Abhängig von den Ergebnissen der Bewertung einer Aufgabe kann definiert werden, welche Folgeaufgaben nach der Bearbeitung einer Aufgabe freigeschaltet werden. Nach einer schlecht bearbeiteten Aufgabe kann eine einfachere Aufgabe aktiviert werden oder bei einer guten Lösung eine anspruchsvollere Aufgabe zur Bearbeitung freigeschaltet werden.
- Optionalität: Für eine Aufgabe kann es mehrere Teilaufgaben geben, welche entweder vollständig oder teilweise (2 von 3) gelöst werden müssen. Auch optionale Teilaufgaben können formuliert werden.

Diese Aufgabenbeziehungen können beliebig miteinander kombiniert werden. Eine Teilaufgabe könnte wiederum selbst aus Teilaufgaben bestehen etc.

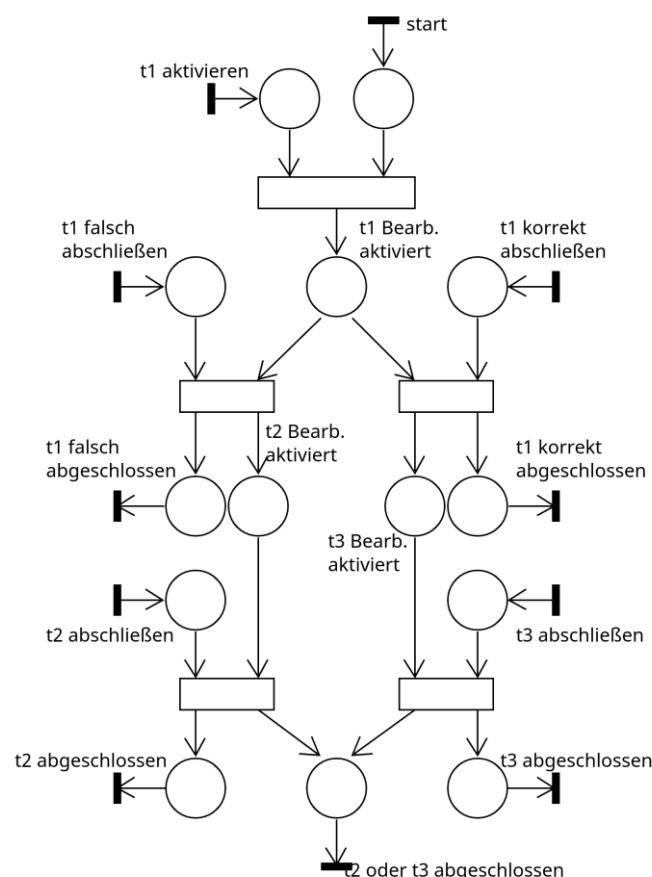


Abbildung 3: Petri-Netz mit bedingter Aufgabensequenz bestehend aus den drei Teilaufgaben t1, t2 und t3. Dabei muss zunächst Teilaufgabe t1 komplett bearbeitet sein, bevor die anderen beiden Teilaufgaben zur Bearbeitung freigegeben werden. Zum Abschluss dieser Aufgabe müssen die Lernenden entweder Teilaufgabe t2 oder Teilaufgabe t3 lösen.

Diese per DSL-Eingabe definierten Aufgabebeziehungen werden in ein Petri-Netz übersetzt, welches zur Spiellaufzeit den Zustand aller Aufgaben überwacht und basierend auf den bearbeiteten Aufgaben weitere Aufgaben zur Bearbeitung freischaltet. Ein Beispiel bestehend aus drei Teilaufgaben zeigt Abb. 3.

5 Grenzen und Ausblick

Das Projekt richtet sich an Lernende mit Interesse an Videospiele. Lernende mit wenig Interesse an Spielen werden nicht motiviert. Das Projekt versteht sich daher als Zusatz zu klassischen Übungsaufgaben. Ansätze zum Einsatz in Prüfungsszenarien wurden untersucht, jedoch muss bedacht werden, dass Studierende in Prüfungssituationen die begrenzte Zeit nicht mit dem Herumlaufen im Level oder dem Besiegen von Monstern verbringen sollten.

Bisher konnte das Projekt noch nicht evaluiert werden. Es soll jedoch eine Evaluierung mit Studierenden und Lehrenden umgesetzt werden. Dabei soll zum einen überprüft werden, ob das Projekt den Lernerfolg der Studierenden fördert als auch ob die DSL für Lehrende gut anwendbar ist und sich die gewünschten Aufgabentypen damit abbilden lassen.

Um den Lernenden eine Hilfestellung zu bieten, falls diese bei der Lösung einer Aufgabe nicht weiterkommen, kann ein dynamisches Hilfesystem implementiert werden. Mithilfe der Petri-Netze könnte dieses den aktuellen Bearbeitungszustand analysieren und gezielte Hilfestellung durch vorher definierte Hilfetexte anbieten.

Aktuell ist das Projekt auf wenige Aufgabentypen begrenzt, was es unter Umständen nicht erlaubt einige analoge Aufgabentypen in das Spiel zu laden.

Nach Bartle [7] gibt es verschiedene Spielertypen. Eine Erweiterung des Spiels um eine Analyse des Spielverhaltens, um den Spielertypen zu bestimmen, würde es erlauben, die Szenarien gezielt nach dem Spielertypen auszuwählen und so den Spaßfaktor für die Lernenden zu erhöhen.

Das QTI-Format [8] ist ein standardisiertes Format, um Aufgaben zu speichern, welches in LMS wie beispielsweise ILIAS eingesetzt wird. Die Entwicklung eines Tools zur Übertragung von QTI-Aufgaben in die DSL würde die schnelle und bequeme Integration bereits bestehender Aufgaben ermöglichen. Dadurch würde die Einstiegshürde gesenkt.

6 Literaturverzeichnis

- [1] Z. C. Schreuders und E. Butterfield, „Gamification for Teaching and Learning Computer Security in Higher Education“, in *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, 2016.
- [2] J. G. Tullis und A. S. Benjamin, „On the effectiveness of self-paced learning“, *Journal of memory and language*, Bd. 64, Nr. 2, S. 109–118, 2011.

- [1] Z. C. Schreuders und E. Butterfield, „Gamification for Teaching and Learning Computer Security in Higher Education“, in *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, 2016.
- [3] J. G. Tullis und A. S. Benjamin, „On the effectiveness of self-paced learning“, *Journal of memory and language*, Bd. 64, Nr. 2, S. 109–118, 2011.
- [4] M. Jurgelaitis, L. Čeponienė, J. Čeponis, und V. Drungilas, „Implementing gamification in a university-level UML modeling course: A case study“, *Computer Applications in Engineering Education*, Bd. 27, Nr. 2, S. 332–343, 2019.
- [5] J. Frentz, M. Tuchscherer, und C. Wiepcke, „Gamified Sustainable Entrepreneurship Education—A digital Educational Escape Room for economy classes in German High Schools“, *STARTPLAY 2022*, S. 30–42, 2022.
- [6] B. Lemoine, P. Laforcade, und S. George, „Mapping Task Types and Gameplay Categories in the Context of Declarative Knowledge Training“, in *15th International Conference on Computer Supported Education*, SCITEPRESS-Science; Technology Publications, 2023, S. 264–275.
- [7] R. A. Bartle, „Player types“, *Jeannie Novak: Game Development Essentials*, Nr. 1, S. 39, 2008.
- [8] S. Lay, „Question and test interoperability: introducing version 2 of the IMS QTI specification“, 2004.

7 Vita

André Matutat, studierte Informatik an der Hochschule Bielefeld und ist seit Oktober 2021 als wissenschaftlicher Mitarbeiter am Fachbereich Campus Minden der Hochschule Bielefeld tätig.

Malte Reinsch, studiert aktuell Informatik im Master an der Hochschule Bielefeld. Er ist seit September 2022 als wissenschaftlicher Mitarbeiter am Fachbereich Campus Minden der Hochschule Bielefeld tätig.

André Kirsch, studierte Informatik an der Hochschule Bielefeld und ist seit Oktober 2021 als wissenschaftlicher Mitarbeiter am Fachbereich Campus Minden der Hochschule Bielefeld tätig.

Dipl.-Inform. Birgit Christina George, erlangte 1990 ihr Informatik-Diplom mit dem Vertiefungsgebiet Compilerbau an der Universität Paderborn. Seit 2012 ist sie als Lehrkraft für besondere Aufgaben im Studiengang Informatik an der Hochschule Bielefeld tätig.

Prof. Dr.-Ing., Dipl.-Ing. Carsten Gips, promovierte 2007 zum Thema "Anwendung von Verfahren des Maschinellen Lernens und von Evolutionären Algorithmen bei der räumlichen Inferenz" an der TU Berlin. Seit 2011 ist er Professor für Programmiermethodik an der Hochschule Bielefeld.