

SimpQui - ein simples Quiz

Stephan Euler TH-Mittelhessen

Version 1.2

12. August 2022

1 Einleitung

In diesem Dokument wird SimpQui – ein simples Quiz – vorgestellt. Dabei stellt sich natürlich sofort die Frage: Warum noch ein Quiz? Quiz gehören zu den ältesten Elementen im eLearning. Es gibt dazu diverse Implementierungen und Untersuchungen zum Nutzen des Einsatzes und aktuelle Lern-Management-Systeme (LMS) beinhalten in der Regel bereits eine entsprechende Komponente. (siehe z. B. [CCF13] [Kap09] [AF13] [RKR20]) [RPW21] [?]). SimQui ist auch keineswegs das Ergebnis eines zielgerichteten Projektes. Vielmehr entstand die Idee ursprünglich als Anwendungsbeispiel in einer Vorlesung zu Java und objektorientierter Programmierung. Basierend auf dem Konzept *Board of Symbols* [Eul18] wurde dabei eine kleine Quiz-Anwendung realisiert. Klassen für verschiedene Fragetypen dienten dabei als Anwendungsbeispiel einer Vererbungshierarchie ausgehend von einer abstrakten Klasse `Question`. Die Fragen und Lösungen standen in einer CSV-Datei und anhand des Parsens der einzelnen Zeilen wurden Methoden zur Verarbeitung von Zeichenketten besprochen.

Das Quiz bewährte sich auch als Unterstützung beim Lernen und die Anzahl der Fragen wuchs. Allerdings wurde in dieser Version eine lokale Kopie der Java-Anwendung benötigt. Außerdem mussten neue Fragen explizit vom Server kopiert werden. Um diese Hürden abzubauen entstand eine Web-Version.

2 Konzept

Das Quiz ist im wesentlichen zum Prüfen des eigenen Verständnisses gedacht. Nach der Vorlesung oder der Bearbeitung von Übungsaufgaben kann man im Quiz zugehörige Fragen versuchen. Im Idealfall kann man die Fragen flüssig richtig beantworten. Umgekehrt sind Schwierigkeiten beim Beantworten ein klarer Hinweis auf Lücken im Wissen oder im Verständnis. In einfachen Fällen – trivialerweise bei Ja-Nein-Fragen – kann man die Lösung durch mehrfaches Probieren erraten. Das kann bereits ein Erkenntnisgewinn sein. Beispiele sind Fragen nach Fakten

wie Jahreszahlen oder Namen. Oft wird es aber auch so sein, dass man dann zwar die richtige Antwort kennt aber noch nicht versteht. Grundlegende Voraussetzung für diesen gedachten Einsatz ist die einfache Bedienung wie im Beispiel: Auf der Heimfahrt in der S-Bahn öffnet ich auf meinem Smartphone die Webseite, wähle den Fragensatz zur besuchten Vorlesung und versuche einige Fragen. Insgesamt orientiert sich die Umsetzung an den folgenden Grundgedanken:

- Möglichst einfach zu verwenden (kein Login), sowohl für Teilnehmer als auch Verwalter
- Diverse Fragetypen, Möglichkeit für dynamische Inhalte
- Direkte Rückmeldung zur Korrektheit der Antwort
- Freie Auswahl der Fragen, keine feste Reihenfolge.
- Spielerische Bedienung, aber keine Gamifikation
- Durch responsive Design auch auf Smartphones verwendbar

3 Umsetzung

3.1 Fragetypen

Jede Frage (eine Ausnahme kommt später) wird durch eine Zeile in einer Fragendatei beschrieben. Dabei dient das #-Zeichen als Trennzeichen. Ein Beispiel ist

```
Open#Wert m nach <pre>for( m=1; m<15; m*=2);</pre>#16
```

Endet eine Zeile mit der Zeichenfolge \\, so wird die nächste Zeile als Fortsetzungszeile gesehen und angehängt. Der allgemeine Aufbau einer Zeile ist

- Typ der Frage
- Die Frage, wobei der Text direkt übernommen wird inklusive eventueller HTML-Auszeichnungen.
- Die Antwort. Im Beispiel mit einer offenen Frage ist dies die gesuchte Lösung. Bei anderen Fragetypen kann es z.B. auch der Index der Lösung unter mehreren Alternativen sein. In diesem Fall folgen die angebotenen Alternative, jeweils wieder durch #-Zeichen getrennt.

Soll ein #-Zeichen im Text erscheinen, so kann man es als HTML-Entität `#` eingeben. Im folgenden sind die bisher implementierten Fragetypen jeweils mit einer kurzen Beschreibung und einem Beispiel aufgeführt.

- YesNo: Eine direkte Frage, die mit *Ja* oder *Nein* beantwortet werden kann.
YesNo#Java ist OOP#true
- Open: eine offene Frage, bei der die Antwort als Text eingegeben wird. Der Text wird auf exakte Übereinstimmung mit der vorgegebenen Antwort geprüft. Lediglich werden eventuell vorhandenen Leerzeichen am Anfang oder Ende entfernt (Methode `trim()`).
Open#i nach <pre>i=4; ++i;</pre>#5
- RegOpen: wie Open, allerdings wird die Antwort als regulärer Ausdruck geprüft. Dadurch kann man beispielsweise Alternativen erlauben.

```
RegOpen#
Wie kann man den Wert <em>eine Million</em>
mit drei Zeichen als Java-Literal darstellen?#
~1[Ee]6$
```

- Correct: Eine Antwort ist vorgegeben. Allerdings müssen noch Fehler korrigiert oder Lücken ausgefüllt werden. Um mehr Freiheit bei der Korrektur zu erlauben, wird die richtige Antwort als regulärer Ausdruck angegeben. Dadurch werden im Beispiel Alternativen (& oder &&) sowie optionale Leerzeichen akzeptiert.
Correct#if(50 <= p <= 100)#if\ (50 <= p +&&? +p +<= 100 \)
- Options: Mehrere Antworten sind vorgegeben und eine davon ist korrekt.
Options#Wie beginnt ein Zeilen-Kommentar in Java?#1#%#/#\#**
- MultiOptions: Eine Verallgemeinerung von Options, bei der mehrere Antworten zutreffen können.
MultiOptions#Was sind gültige int-Werte?#0,2#007#009#011
Die Anzahl der richtigen Optionen ist nicht beschränkt. Es ist also auch möglich, eine Frage zu stellen, bei der keine der angebotenen Optionen zutrifft.
- Order: Hier ist die Aufgabe, die vorgegebenen Antworten in die richtige Reihenfolge zu bringen. Die Reihenfolge soll in der Art ADCB angegeben werden. Die Prüfung ist recht tolerant: alle Nicht-Buchstaben werden aus der Antwort entfernt und Buchstaben werden auf Großschreibung gewandelt. Damit werden auch Antworten wie adcb, A D C B oder A-d-C-B erkannt.

```
Order#
Ordnen Sie die Modifikatoren nach zunehmender Sichtbarkeit#
ABC#private#protected#public
```

- Dyn: mit entsprechenden PHP-Methoden können Fragen dynamisch generiert werden. Bei diesem Fragetyp wird der Name einer entsprechenden Methode angegeben. Die Methode soll ein Frage-Objekt zurückgeben. Als Beispiel wird bei

Dyn#mod

die Methode

```
static function mod() {
    $a = random_int(12,30);
    $b = random_int(2, 7);
    return new OpenQuestion( "$a_%$b", $a % $b );
}
```

ausgeführt. Hier werden zwei Zufallswerte aus den spezifizierten Bereichen erzeugt. Aus den beiden Werte werden dann die Fragestellung und die Lösung erstellt und im Konstruktor für einen `OpenQuestion` übergeben.

- Feedback: Mit diesem Sondertyp kann eine einfache Feedback-Möglichkeit in ein Quiz eingebunden werden. Vorgegeben ist eine Frage. Die Antwort wird dann mit einer speziellen Kennung in die Protokoll-Datei geschrieben. Aktuell ist eine Version mit einer Bewertung auf der Skala 1 bis 6 implementiert.

Feedback#Wie beurteilen Sie das GUI (in Schulnoten 1 bis 6)?

- FeedbackOpen: Hier kann ein kurzer Text eingegeben werden.

Allerdings sind die Fragetypen nicht streng voneinander getrennt. So kann man ein YesNo-Frage auch als Options-Frage mit zwei Auswahlmöglichkeiten realisieren. Eine Options-Frage wiederum kann als MultiOptions-Frage mit nur einer korrekten Lösung gesehen werden. In solchen Fällen kann man den Fragetyp je nach gewünschter Darstellung in der Webseite wählen.

Nun noch zur angesprochenen Ausnahme: In der Programmierausbildung verwendet man oft Aufgaben in der Art „Welche Ausgabe erzeugt der folgende Code?“. Dabei wird ein mehr oder weniger umfangreicher Code-Abschnitt vorgegeben. Als Antwort wird die produzierte Ausgabe gesucht. Ein erste Implementierung solcher Frage ist wie im folgenden Beispiel möglich:

```
<<
for( int i=0; i<30; i+=10 ) {
    for( int j=0; j<3; j++ ) {
        System.out.print( i + j + " ");
    }
}
```

```
#
0 1 2 10 11 12 20 21 22
>>
```

Die Frage mit Antwort umfasst mehrere Zeilen und wird durch « und » begrenzt. Derzeit sind nur einzeilige Antworten möglich.

3.2 Dateien

Die Fragen sind in Dateien gesammelt. Erwartet wird die Dateierweiterung `.txt` und eine Datei kann neben den Fragen Kommentarzeilen (beginnend mit einem `!`-Zeichen) enthalten. Leerzeilen werden ignoriert und können zur Abgrenzung der Fragen eingefügt werden.

Die übergeordnete Einheit sind Topic-Dateien (`.top`). Eine solche Datei enthält Kommentare und eine Liste von Fragesätzen. Im einfachsten Fall wird ein Fragesatz durch die Angabe der entsprechenden Fragedatei (der Pfad ohne die Endung `.txt`) eingebunden. Dieser Name wird dann auch im Menü der Webseite angezeigt. Man kann allerdings auch einen anderen Anzeigenamen vorgeben. Im folgenden Beispiel wird diese Möglichkeit genutzt, um bei der Einheit zu Integer nicht den Pfadnamen mit Unterverzeichnis anzuzeigen.

```
!Vorlesung Programmierung 1 <br>
!Version 16. Juni 2020
Java
Integer:datentypen/Int
Abläufe
datentypen/Float
Print
Felder
Methoden
Mixed:mix | add Int | add Float | random | head 24
---
RegEx
```

Der Eintrag `Mixed` zeigt noch Möglichkeiten zum Kombinieren und Mischen von Fragesätzen. Ausgehend von einem Satz – im Beispiel `mix` – werden nach dem Vorbild der Unix-Pipes Aktionen ausgeführt. Im Beispiel werden zwei weitere Sätze hinzugefügt. Die Gesamtmenge an Fragen wird dann zufällig verwürfelt und daraus wiederum werden die ersten 24 Fragen gewählt. Im Ergebnis werden dann beim Aufruf des Fragesatzes 24 zufällige Fragen aus den drei Sätzen `mix`, `Int` und `Float` gewählt. Angezeigt werden die Kommentare aus dem ersten Satz, der aber ansonsten auch leer sein kann. Eine Zeile `--` definiert – wie im Standard Markdown – eine Trennlinie.

3.3 Hilfen und Erläuterung

Jede Frage kann um zusätzliche Informationen ergänzt werden. Zum einen kann eine Hilfe eingetragen werden, die nach einer vorgegebenen Anzahl von Fehlversuchen angezeigt wird. Zum anderen kann eine *Erläuterung nach richtiger Beantwortung* eingetragen werden. Beide Möglichkeiten werden durch zusätzliche Zeilen nach einer Frage in die Datei geschrieben. Das folgende Beispiel zeigt diese Möglichkeiten:

```
MultiOptions#Was sind gültige int-Werte?#0,2#007#009#011
Hint#1#Was bedeutet eine 0 am Anfang?
Explain#Die folgende 0 kennzeichnet Oktalzahlen. \\
Dann sind aber nur die Ziffern 0 bis 6 erlaubt.
```

Die Hilfe wird in diesem Fall bereits nach einem Fehlversuch angezeigt. Wahlweise kann auch nur eine der Möglichkeiten genutzt werden. Die Reihenfolge der beiden Zeilen spielt keine Rolle. Die zusätzlichen Informationen können durch die Filter

```
noExplain
noHelp
```

ausgeblendet werden. So könnte in einer Topic-Datei testweise der gleiche Fragesatz in zwei verschiedenen Versionen verwendet werden:

```
Beispiel: tests/fragen
BeispielOhne: tests/fragen | noExplain | noHelp
```

3.4 Implementierung

Die Anwendung wurde in den Programmiersprachen PHP und JavaScript implementiert. Der gesamte Code ist auf GitHub eingestellt¹. Der PHP-Teil auf dem Server übernimmt die Navigation zwischen den einzelnen Seiten. Dabei werden aus den Topic-Dateien passende Auswahlmenüs erstellt. Wird ein Fragesatz ausgewählt, so werden die Fragen aus der Datei geladen. Je nach Typ wird für jede Frage passender HTML-Code generiert. Der gesamte Code für den Fragesatz wird dann an den Client geschickt. Die Auswahl der Fragen sowie die Überprüfung der Antworten übernimmt der JavaScript-Teil im Client.

Bild 1 zeigt die Darstellung der Fragen im Browser. Der Satz enthält insgesamt 30 Fragen, aus denen man beliebig auswählen kann. Richtig beantwortet Fragen werden mit einem freundlichen smiley markiert. Bei Fehlversuchen wird der Frageknopf rot gefärbt. Der Client führt eine Statistik der Versuche und der richtigen Antworten. Allerdings bezieht sich diese Auswertung nur auf die aktuelle Seite. Wechselt man zu einem anderen Fragesatz (oder lädt die Seite neu) starten die Zähler wieder bei 0.

¹<https://github.com/stephaneuler/SimpQui>

Fragen in der Einheit *Int*, Thema PG1

Aufgaben zu int - Werten und Variablen

| | | | | | | | | | | | | | | |
|----|----|----|----|---|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 😊 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 😊 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | | |
| 29 | 30 | | | | | | | | | | | | | |

Frage 27

Welche Anweisungen sind erlaubt?

☐ `int n = [2 + 3] * 5;`
☐ `int n = {2 + 3} * 5;`
☐ `int n = (2 + 3) * 5;`

Okay

Antworten: 3, Richtig: 2 (67%)

Abbildung 1: Beispiel Fragensatz

Für allgemeine Auswertungen wird jeder Versuch protokolliert. Bei jedem Versuch wird Datum, Thema, Frage und Ergebnis gespeichert. Da die Fragen keinen Index haben, dient der Fragetext zur Identifikation. Auf der Webseite kann man die letzten 10 Versuche einsehen. Außerdem wird die mittlere Erfolgsquote angegeben (Bild 2).

3.5 Export

Im Sinne einer Weiterverwendung der Fragen ist der Export in andere Formate wünschenswert. In der derzeitigen Version ist bereits eine erste Version eines Konverters zu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ implementiert. So wird aus der Frage

MultiOptions#Was sind gültige int-Werte?#0,2#007#009#011

der Text

```

Was sind gültige int-Werte?
\begin{itemize}
\item[ $\bigcirc$ ] 007
\item[ $\bigcirc$ ] 009
\item[ $\bigcirc$ ] 011
\end{itemize}

```

generiert. Weitere Converter – insbesondere zu den weit verbreiteten Moodle-Formaten wie z. B. Moodle GIFT² – sind in Arbeit.

²<http://microformats.org/wiki/gift>



Abbildung 2: Statistik mit den letzten 10 Versuchen

4 Erste Erfahrungen und Ausblick

Etwas seit Anfang 2020 ist eine Quiz-Webseite³ online. Der Schwerpunkt liegt auf Fragen zu Grundlagen der Programmierung in Java. Mittlerweile wurden etwa 180.000 Versuche gezählt (Stand Januar 2022) bei einer Quote von 62% richtigen Antworten. Es traten keine größeren technische Probleme auf. Lediglich bei der Darstellung von Auswahlfragen auf Smartphones wurde eine Schaltfläche als unnötig kritisiert. Allerdings wird diese wohl vom Betriebssystem generiert und bisher wurde noch keine Möglichkeit gefunden, sie auszublenden.

Die Weiterentwicklung der Anwendung ist im Rahmen zukünftiger Lehrveranstaltungen geplant. Beispielsweise könnten weitere Fragetypen ergänzt werden. Hierbei sind insbesondere über Textfragen hinausgehende Möglichkeiten interessant. Als erster Prototyp ist bereits die Aufgabe *Skizzieren Sie den Verlauf der Funktion x^2* realisiert. Dabei soll auf einem Canvas-Element der Funktionsverlauf eingezeichnet werden. Anhand des Abstandes zwischen dieser Skizze und der abgespeicherten Lösung wird dann auf ausreichende Übereinstimmung geprüft.

Zur Auswertung der Versuche der Protokolldatei erfolgt derzeit mit einer eigenen Java-Anwendung. Dann lassen sich beispielsweise Fragen mit hoher Fehlerquote identifizieren. Die Auswertung könnte alternativ in die Webseite integriert werden, um eine direkte Auswertung zu erlauben.

³<https://hosting.iem.thm.de/user/euler/quiz/index.php>

Literatur

- [AF13] Giovannina Albano and Pier Luigi Ferrari. Mathematics education and e-learning: meaningful use of the quiz-modules. In *Proceedings of ICTMT11 (International Conference on Technology in Mathematics Teaching)*, pages 53–58, 2013.
- [CCF13] Christopher Cheong, France Cheong, and Justin Filippou. Quick quiz: A gamified approach for enhancing learning. In *PACIS*, page 206, 2013.
- [Eul18] Stephan Euler. Board of symbols: A tool for teaching programming. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018, pages 352–353, New York, NY, USA, 2018. ACM.
- [Kap09] Vassilis Kapsalis. Implementation of an assessment system incorporating web-based parameterized questions. *International Journal of Emerging Technologies in Learning (iJET)*, 4(3):20–28, 2009.
- [OF13] Rebecca Orr and Shellene Foster. Increasing student success using online quizzing in introductory (majors) biology. *CBE—Life Sciences Education*, 12(3):509–514, 2013.
- [RKR20] C Daniel Riggs, Sohee Kang, and Olivia Rennie. Positive impact of multiple-choice question authoring and regular quiz participation on student learning. *CBE—Life Sciences Education*, 19(2):ar16, 2020.
- [RPW21] Irene Rothe, Willi Pustowalow, and Marco Winzker. Multipurpose use of quizzes in teaching. In *2021 IEEE Global Engineering Education Conference (EDUCON)*, pages 63–67, 2021.