

Dokumentation

JavaQuiz 24.04.2021

Vorname	Nachname	E-Mail	Matrikelnummer
Mohamed Lemine	Mohamed El Agheb	mohamed-lemine.mohamed-el-agheb@stud.hshl.de	2170193
Karuan	Osi	karuan.osi@stud.hshl.de	2180189
Hakan	Koyun	hakan.koyun@stud.hshl.de	2130672

1. Protokollierung der Tätigkeiten und Aufgabenverteilung

Datum	Tätigkeit
24.04.2021	Die Gruppe hat sich gemeinsam für ein Spiel entschieden, welches programmiert werden soll, und zwar: JavaQuiz. Lemine hat der Gruppe erklärt, wie man GitHub und Git richtig verwendet. Karuan hat sich Fragen für das Quiz überlegt
30.04.2021	Lemine hat die Klasse QuizView erstellt und dran gearbeitet. Einpfelegen der Fragen und Auswahlmöglichkeiten in die Klasse Question.
07.05.2021	Lemine hat weiter an der GUI gearbeitet und versucht in den Vorlesungen zu GUI Gelerntes umzusetzen. Karuan hat die Klasse QuizController und QuizModel erstellt und hat sich Gedanken darüber gemacht, wie das Model View Controller Muster richtig eingesetzt werden kann.
14.05.2021	Hakan hat an der Klasse QuizController weitergearbeitet. Karuan haben an der Klasse QuizModel gearbeitet.
21.05.2021	Probleme: Das Spiel funktioniert mit MVC nicht wie erwartet. Es wird von den Teammitgliedern nach einer Lösung gesucht.
28.05.2021	Lemine und Hakan haben ein Package erstellt (package: quiz), in dem eine andere Version des Spiels implementiert wird (Ohne MVC), damit eine funktionierende Version vorliegt.
04.06.2021	Das Team hat zusammen eine Lösung gefunden (Probleme gefunden und gelöst)) und das Programm funktioniert wie gewünscht war mit dem MVC Muster.
11.06.2021	Hakan hat sich mit den Folien zu verteilten Systemen auseinandergesetzt und versucht, das Spiel verteilt zu implementieren.

1. Einleitung

Worum geht es?

JavaQuiz ist ein Projekt, das ein einfaches Prinzip verfolgt. Es ist ein Programm, bei dem 10 Fragen rund um die Programmiersprache Java beantwortet werden sollen. Für jede Frage hat der Spieler 15 Sekunden Zeit. Sollte eine Frage nicht vor dem Ablauf der 15 Sekunden beantwortet worden sein, wird sie als falsch gewertet und es geht automatisch zur nächsten Frage weiter.

Sobald die letzte Frage beantwortet wurde, erfährt der Spieler wieviel Fragen richtig beantwortet wurden und was der Durchschnitt der richtig beantworteten Fragen im Allgemeinen ist.

2. Produktbeschreibung

Beschreiben Sie die Ziele des Spiels und den Produkteinsatz.

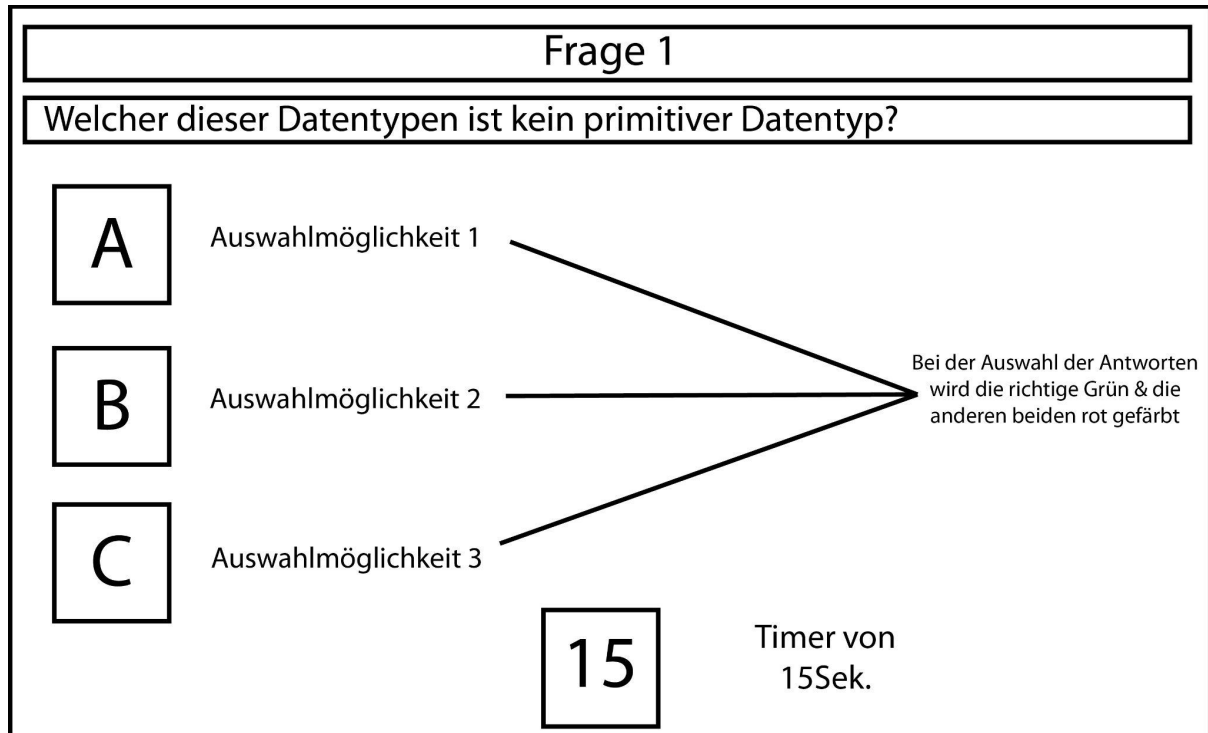
Bei JavaQuiz wird dir eine Frage angezeigt und man kann aus drei Antwortmöglichkeiten, eine Antwort mit der Maus angeklickt werden. Der Timer zählt von 15 runter. Sobald eine Antwort angeklickt wurde, färbt sich das angeklickte Feld entweder Grün (richtige Antwort) oder das Feld färbt sich Rot (Falsche Antwort). Es gibt 10 Fragen. Visualisiert ist das Spiel durch eine grafische Oberfläche.

Ziel ist es, in der gegebenen Zeit so viele Antworten wie möglich richtig zu beantworten.

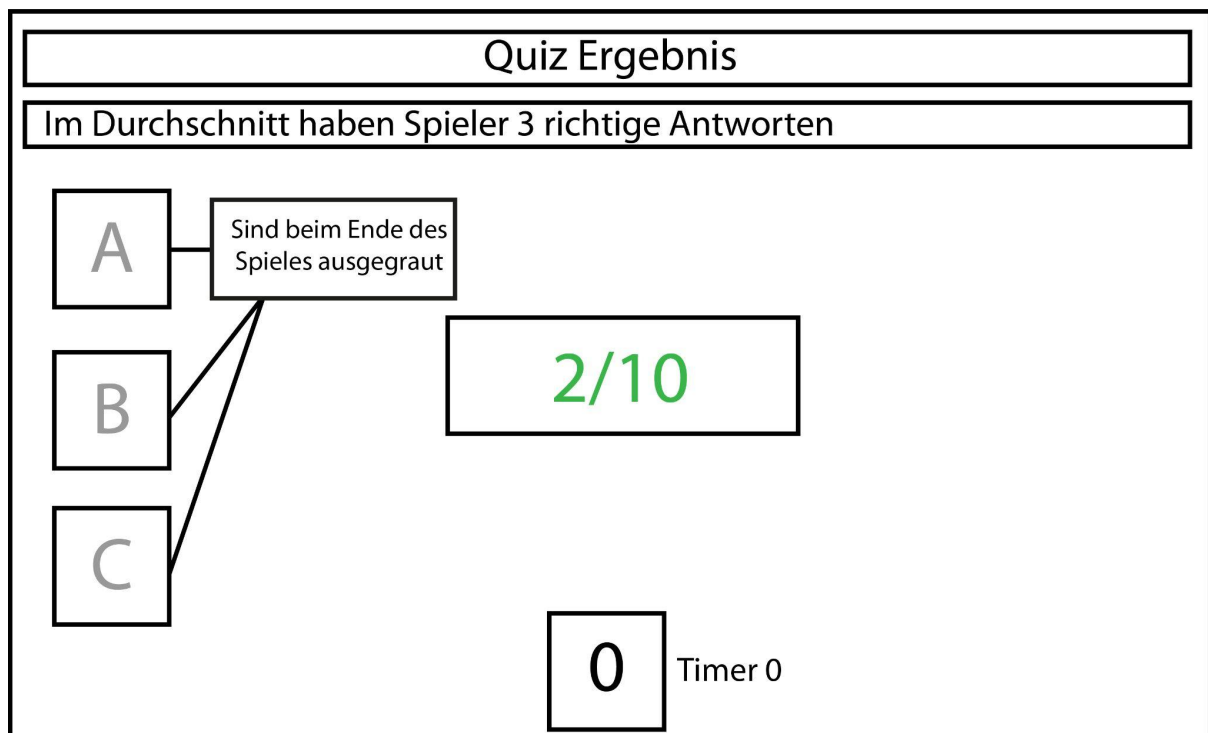
Mit dem Spiel kann man sein Wissen auf die Probe stellen. Es ist auch möglich das Spiel zu zweit zu spielen und gegeneinander anzutreten. Dafür wird auf unterschiedlichen Rechnern gespielt und am Ende können die Ergebnisse verglichen werden, um zu sehen wer die höchste Punktzahl erzielt hat. Der/Diejenige mit der höchsten Punktzahl gewinnt.

3. Skizzen

Startscreen



Endscreen



4. Aufbau der Software

Package mvc (Model View Controller)

Klassenname: Main

- Startet das Programm, um das Spiel (JavaQuiz) spielen zu können.

Klassenname: QuizController

- ggf. Prüfungen der Spielstände und Antworten etc.

Klassenname: QuizModell

- Spiellogik

Klassen-Name: QuizView

- für die graphische Benutzeroberfläche zuständig.

Package quiz

Klassenname: QuizModelAndView

- Spiellogik und GUI

Klassen-Name: Question

- Enthält alle Fragen und Antwortmöglichkeiten in Arrays

Klassen-Name: Storage

- erzeugt eine txt. Datei und speichert dort die Anzahl der richtig beantworteten Fragen mit dem Datum & Uhrzeit, an dem das Spiel beendet wurde ein.
- Beim Spielende wird die Datei ausgelesen und der Durchschnitt der richtig beantworteten Fragen berechnet.

5. Voraussetzungen

Voraussetzungen, die erfüllt werden müssen, um das Programm starten zu können:

Um die Software ausführen zu können, muss folgendes vorhanden sein:

- Eine aktuelle Java Version (mindestens Java 11)
- Entweder im Terminal (Mac) oder CMD/PowerShell (Windows) den Befehl ***java -jar javaquiz.jar*** eingeben und die Eingabe bestätigen, um die Jar-Datei auszuführen, um damit das Spiel zu starten.

6. Positiver / Negativer Ereignisse

Positive Ereignisse	Negative Ereignisse
Unser Team weist eine sehr gute Gruppendynamik auf	Probleme bei der Umsetzung des Model View Controllers
Mehrere Meetings haben geholfen, um bestimmte Probleme von anderen Teammitgliedern zu lösen	Viele kleine Bugs, die auf den ersten Blicken nicht zu erkennen waren. Dies hat uns sehr viel Zeit gekostet und uns bei unserem Zeitplan durcheinander gebracht.
Aufgabenverteilung für die nächsten Wochen wurden rasch vergeben.	Schwierigkeit bei der Implementierung verteilter Programmierung