

Programmierung eines Vocitrainers

Untertitel

Inhaltsverzeichnis

Einleitung	3
Anforderungen und Zielgruppe	3
Aufbau	4
Client-Server-Modell	4
Datenspeicherung	4
Wie lerne ich am Besten	5
Wiederholung macht den Meiser	5
Die magische Zahl 7 ± 2	5
Pomodoro-Technik.....	6
Was man schreibt, das bleibt	6
Qt-Framework	6
Was ist ein Framework.....	6
Funktion	6
PyQt	6
Beispiel eines Miniprogrammes.....	6
Signale und Slots	6
Objekt Orientierte Programmierung	6
QtDesigner	6
QtLinguist.....	6
Bibliotheken und Tools.....	6
Pythons Standartbibliothek.....	6
PyCharm.....	6
Git.....	6
ChatGPT	6
Dokumentation.....	7

Einleitung

Man kann dem Umstand nicht aus dem Weg gehen, Vokabeln lernen gehört zum Alltag eines jeden Schülers. Niemand macht es gerne, aber man muss durch. Da stellt sich die Frage, wie man mit möglichst wenig Zeitaufwand seinen Wortschatz effizient und nachhaltig erweitern kann.

Früher haben wir vermutlich alle noch mit analogen Karten aus Papier und einem Karteikasten gelernt. Mit der Zeit haben die meisten Quizlet für sich entdeckt. Quizlet macht vieles leichter. Der Vorteil dieser Webseite ist es, dass man sich bereits von anderen Benutzern erstellte Lernsets von Wörtern herunterladen kann und dabei fast jedes Lehrmittel auf der Seite abgedeckt ist. Auch sonst muss nur eine Person das Set erstellen und kann es danach mit allen anderen teilen.

Quizlet war übersichtlich, hatte alle nötigen Funktionen ohne Schnick-Schnack, es war alles was man brauchte. Leider liegt hier die Betonung auf «war». Wie es so häufig ist, sobald ein Produkt beliebt ist, fängt das dahinterstehende Unternehmen an, den Gewinn zu maximieren. So waren mit der Zeit immer mehr Funktionen nur noch mit Quizlet Plus, der bezahlten Version von Quizlet, verfügbar. Aber vor allem wurde Quizlet unübersichtlich und voller Ablenkungen. Es gibt unnötige Lernspiele, nervige Werbung und KI-Assistenzen, welche vom eigentlichen Kern ablenken: Möglichst zeitsparend Vokabeln lernen.

Das hat mich bereits vor 2 Jahren aufgeregt. Ich habe damals gerade gelernt mit Office-Makros, das sind kleine Programme in einem Office Dokument, zu programmieren. Deshalb habe ich direkt in der Excel-Datei, in der wir die Englisch-Wörter bekamen, begonnen ein Vokabeltrainer zu programmieren. Das Projekt wuchs und der «Vocitrainer» unterstützt inzwischen fünf Sprachen, bietet drei Lernmodi an und hilft mir mit der Aussprache des Wortes direkt im Programm.

Allerdings hat dieser Vokabeltrainer aus verschiedenen Gründen, wie einer mangelhaften Wartung von Seite Microsofts, keine grosse Zukunft. Ich habe immer wieder mit dem Gedanken gespielt, den «Vocitrainer» in der inzwischen erlernten Programmiersprache 'Python' von Grund auf neu zu schreiben. Als ich in der sechsten Klasse begonnen habe, mir Gedanken über meine Maturaarbeit zu machen, kam schnell die Idee auf, diese Gedanken in die Tat umzusetzen.

Anforderungen und Zielgruppe

Da meine Ressourcen für das Projekt begrenzt sind, ist es wichtig, klare Anforderungen zu setzen und andere «unnötige» Funktionen wegzulassen.

1 Mir ist es wichtig, dass es mit dem neuen Vokabeltrainer möglich ist, ganz unkompliziert selbst erstellte Lernsets mit anderen zu teilen, indem man sie auf einen öffentlichen 'Marketplace' hochlädt. Jeder andere Benutzer kann diesen durchsuchen und sich Lernsets herunterladen.

2 Da aber nicht von Anfang an viele Lernsets bereits zu Verfügung stehen werden, ist es mir wichtig, dass man ganz einfach die Lernsets, welche man zum Beispiel in Form von CSV-Dateien schon hat, importieren kann. Optimal wäre ein Tool, dass Lernsets direkt von Quizlet herunterlädt.

3 Die wichtigste Eigenschaft eines Vokabeltrainers ist es, dass man damit effizient lernen kann. Gerade weil man in der Praxis nicht regelmässig lernt sondern erst die paar Tage vor der Prüfung, ist es wichtig, dass der Algorithmus ein schnelles Lernen fördert. Deshalb versuche ich, Erkenntnisse aus der Lernpsychologie und der Funktionsweise des Gehirns direkt in den Abfragealgorithmus einzubauen.

4 Meine Erfahrung ist es, dass Webapplikationen zur Ablenkung verführen, da sie im Browser offen sind und damit nur ein bis zwei Klicks von den neusten Nachrichten und kleinen Onlinespielen entfernt sind. Da Ablenkung für das Gehirn den Lerneffekt massiv reduzieren, gilt es das Potenzial dafür möglichst zu minimieren. Deshalb und weil ich bereits Python kann, habe ich mich entschieden, dass Programm standalone und damit offline nutzbar zu programmieren.

Die Zielgruppe sind Benutzer, die sich nicht mit den üblichen Lösungen zufrieden geben, Ablenkung minimieren und Effizienz maximieren wollen: Der typische computeraffine «Poweruser».

Aufbau

Client-Server-Modell

Das Client-Server-Modell ist ein Konzept aus der Informatik, um Aufgaben aufzuteilen. Der Server ist ein Computer, der wie ein Dienstleister fungiert. Er hat Informationen, welche von jedem Kunden (Client) angefordert werden können.

Dieses Konzept hilft mir dabei, einen Marketplace, wie in den Anforderungen beschrieben, umsetzen zu können. Die Clients sind dabei die Schüler, mit ihrem Programm. Wenn nun beispielsweise eine Schülerin ein neues Set von Vokabeln mit ihren Mitschülern teilen will, lädt ihr Programm dieses im Hintergrund auf den Server. Der Server-Computer speichert das auf seiner Festplatte ab. Wenn jetzt die anderen Mitschüler das Set herunterladen wollen, schickt ihr Programm im Hintergrund eine Anfrage «Bitte schicke mir das Set 'xy'» an den Server. Dieser antwortet mit dem Set, welches dann im Vokabel Trainer angezeigt wird.

Datenspeicherung

Bei einem Vokabel Trainer gibt es viele Daten, welche gespeichert werden müssen. Dabei müssen sowohl Daten auf dem Client, wie auch Daten auf dem Server gespeichert werden. Beim Benutzer auf dem PC werden die ganzen Vokabeln und Sets gespeichert, sowie die Ordner Struktur (welches Lernset in welchem Ordner ist). Der Server muss sich zusätzlich noch die verschiedenen Benutzer und ihre Passwörter merken.

Es gibt verschiedene Formate, um Dateien zu speichern. Ich könnte die Daten mit einer eigenen Struktur in einer einfachen Textdatei speichern. Eine alternative dazu wären Formate wie JSON-Dateien und XML-Dateien, welche bereits eine definierte Struktur haben.

Ich habe mich für eine Datenbank entschieden. In einer Datenbank kann man auch Daten strukturiert speichern, aber sie bietet noch einige Vorteile.

1. In einer Datenbank werden Daten in einer fixen Struktur abgelegt. Das gewährt die Konsistenz und Integrität der Daten: Daten *können nicht* «falsch» gespeichert werden.
2. Mit der Abfragesprache SQL kann ich Daten aus der Datenbank einfach sortieren oder nach bestimmten Datensätzen suchen.
3. Datenbanken sind für grosse Datenmengen und viele Benutzer optimiert und verbrauchen deshalb weniger Ressourcen.

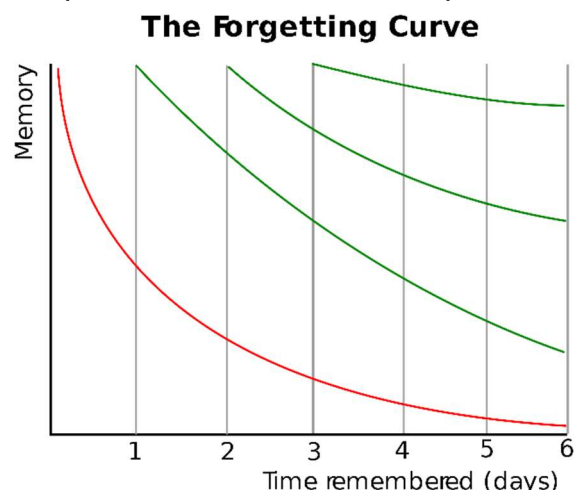
Stark vereinfacht kann man sie Datenbanken wie eine grosses Excel-Dokument vorstellen. Wie auch in einem solchen Dokument, gibt es bei einer Datenbank unterschiedlicher Tabellen, in welchen verschiedenartige Informationen gespeichert werden. In der Client Datenbank gibt es die Tabellen «karte», die Tabelle, «vociset» und die Tabelle «ordner».

Auf dem Server gibt es ebenfalls «karte» und «vociset». Da es auf dem Server keine hierarchische Ordnerstruktur gibt, in der die Vokabel Sets gespeichert werden, braucht es keine Tabelle «ordner.» Dafür gibt es eine Tabelle «user», in welcher die verschiedenen Benutzer mit E-Mail, Passwort und Benutzername gespeichert werden.

Wie lerne ich am Besten

Wiederholung macht den Meiser

Ende des 19. Jahrhundert untersuchte der deutsche Psychologe Herman Ebbinghaus den Einfluss von Wiederholungen auf den Lerneffekt. Er fand heraus, dass das Lernen effektiver ist, wenn die Lerninhalte über einen längeren Zeitraum verteilt werden. Seine bekannteste Arbeit ist die Vergessenskurve, welche wohl die meisten Menschen schon mal gesehen haben. Die Rote Linie zeigt die Abnahme des verbleibenden Wissens über die Zeit. Wenn man jedoch nach einem Tag die Lerninhalte nochmals wiederholt, vergisst man nicht mehr so schnell (1. Grüne Kurve) Mit jeder Wiederholung wird die Kurve flacher. Nach mehrmaligem Wiederholen sinkt allerdings der Verbesserungseffekt. Überlernen lohnt sich also nicht, da es nur kleine Vorteile bringt.



Die magische Zahl 7 ± 2

Selbstversuch

Schritt 1: Auf der nächsten Seite findest du einen weiteren orangen Kasten mit 20 Wörtern. Präge dir die Wörter gut ein, ohne sie aufzuschreiben! Du hast 1 Minute Zeit.

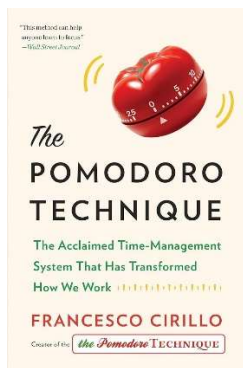
Schritt 2: Schreibe nun ohne nachzuschauen alle Wörter an die du dich erinnern kannst auf ein Blatt Papier. Du hast dafür etwa eine halbe Minute Zeit.

Schritt 3: Zähle, wie viele Wörter du richtig hast. Vermutlich konntest du dich an 5-9 Wörter erinnern. Diese Limitierung ist es, welche «Millers Law» beschreibt.

1956 veröffentlichte George A. Miller seinen berühmten Artikel «The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information». Darin beschreibt er den Umstand, dass wir uns nur ungefähr 7 Informationseinheiten im Kurzzeitgedächtnis merken können.

Diese Erkenntnis hat ganz konkrete Folgen darauf, wie wir am effizientesten neu Wörter lernen. Ein Wort zu erlernen ist ein Prozess. Wenn man es zum ersten Mal sieht, versucht man sich die Bedeutung einzuprägen, beim zweiten bzw. dritten Mal versucht man bereits die richtige Schreibweise zu lernen. Miller's Law angewendet heisst also, dass nur immer 7 Vokabeln auf einmal in diesem Lernprozess sein dürfen. Sind es mehr, nimmt der Lernerfolg rapide ab: Das Wort wird aus dem Kurzzeitgedächtnis verdrängt, bevor es wieder abgefragt und damit gefestigt wird.

Pomodoro-Technik



Häufig fällt es uns schwer, uns zu konzentrieren. Wir leiden unter der Prokrastination, unserem Drang, nur kurz unser Social Media zu checken. Dem Italiener Francesco Cirillo ging es genauso. Deshalb hat er die Pomodoro-Technik entwickelt. Er nahm eine Küchenuhr (in Tomatenform, ital. Pomodoro) und stellte sie auf 10 Minuten, während diesen wollte er voll fokussiert arbeiten. Später hat er die Technik weiter entwickelt und ein Buch darüber geschrieben (siehe Bild).

Die Technik funktioniert folgendermassen:

1. Man stellt sich eine Aufgabe
2. Eine Uhr auf 25 Minuten einstellen
3. 25 Minuten fokussiert durcharbeiten
4. Alles stehen lassen und 5 Minuten Pause machen
5. Das kann man mehrmals wiederholen. Nach vier 'Pomodori' sollte man eine grössere Pause von bis zu 30 Minuten einlegen.

Die Idee ist, dass man mit einem klar definierten Ziel und beschränkter Zeit besser konzentrieren kann, ohne sich ablenken zu lassen.

Natürlich wird das zuerst nicht gelingen, aber das schöne ist, die Pomodoro-Technik lässt sich trainieren. Und dank den klaren Einheiten, lässt sich leicht messen, wie produktiv man war.

Was man schreibt, das bleibt

Eine neuere Studie von 2014 namens «The Pen Is Mightier Than the Keyboard: Advantages of Longhand Over Laptop Note Taking» zeigt, dass das handschriftliche Schreiben Vorteile gegenüber dem Tippen hat. Auch wenn die Studie vor allem untersuchte, wie die Unterschiede beim Notizen machen von Videos sind, lässt sich einiges für das Vokabellernen ableiten.

1. Da das handschriftliche Schreiben länger dauert, hat das Gehirn mehr Zeit, das Geschriebene zu verinnerlichen.
2. Der Akt des Schreibens mit der eigenen Hand hat ein vertieftes Lernen zur Folge
3. Und speziell für das Abschreiben von Vokabeln: Gerade bei schwierigen Wörtern hilft es, dass wir mit dem Blick mehrmals zwischen dem Wort und dem selbst geschriebenen wechseln, was den Lerneffekt weiter verbessert.

Die Frage ist nun, wie sich diese Tatsache in einen digitalen Vokabel-Trainer einbinden lässt. Natürlich kann man es nicht erzwingen, aber wenn ein Wort zum ersten Mal gezeigt wird, zeigt der Vokabeltrainer eine Aufforderung, das Wort einmal handschriftlich abzuschreiben.

Selbstversuch

Haus
Quadrat
Tür
Buch
Apfel
Auto
Schuh
Kugel
Band
Schlüssel
Fisch
Box
Kopf
Hammer
Pfeil
Katze
Blume
Milch
Baum
König

Qt-Framework

Bibliotheken und Tools

Dokumentation

¹ Wikipedia über Qt: [https://de.wikipedia.org/wiki/Qt_\(Bibliothek\)#Verwendungsbeispiele](https://de.wikipedia.org/wiki/Qt_(Bibliothek)#Verwendungsbeispiele)

² PyQt5 Dokumentation: <https://www.riverbankcomputing.com/static/Docs/PyQt5/introduction.html>