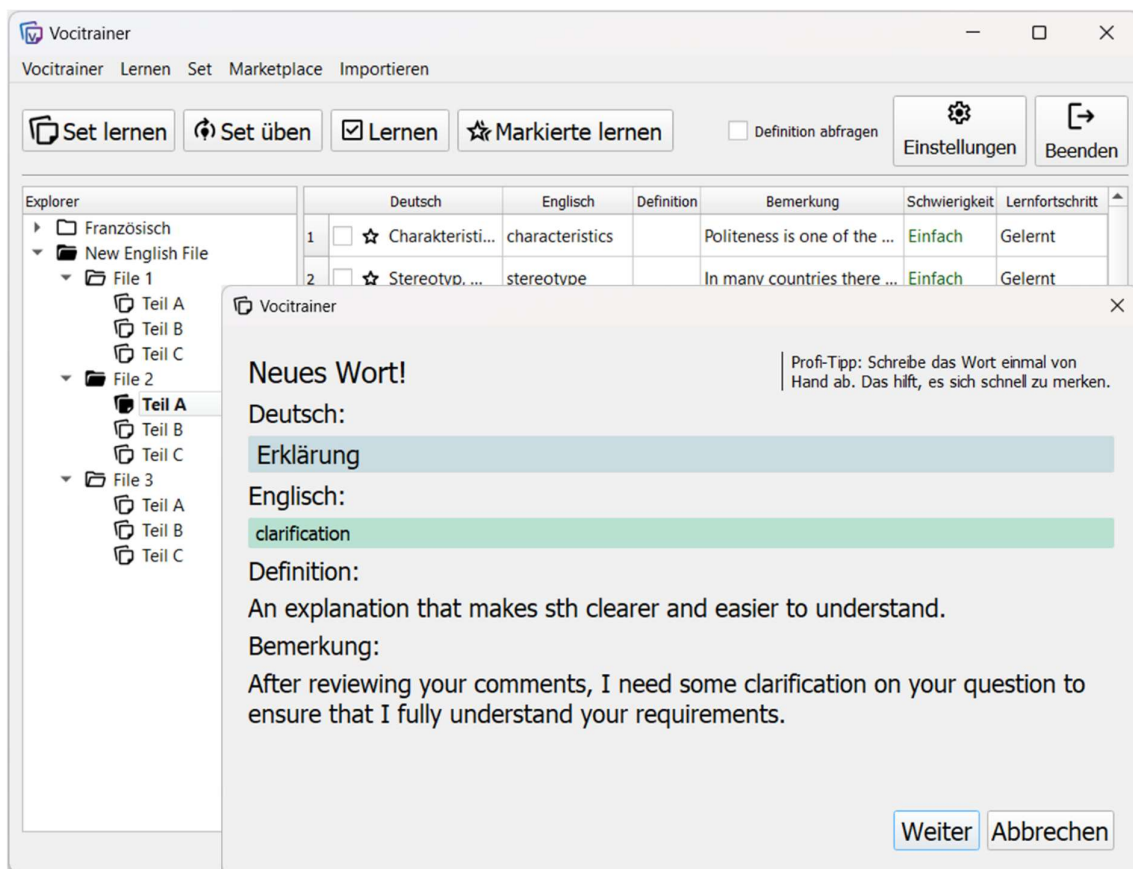


Programmierung eines Vokabeltrainers

Maturaarbeit von Samuel Barmet



betreut von M. Kubba

Inhaltsverzeichnis

Einleitung	4
Anforderungen und Zielgruppe	5
Aufbau	6
Client-Server-Modell	6
Datenspeicherung	6
Wie lerne ich am besten	9
Wiederholung macht den Meister.....	9
Die magische Zahl 7 ± 2	9
Pomodoro-Technik.....	10
Was man schreibt, das bleibt	11
Funktionen	12
Das Hauptfenster.....	13
Das Training	14
Der Marketplace	15
Qt-Framework	16
Was ist ein Framework.....	16
Funktion	16
PyQt	17
Beispiel eines Miniprogrammes	17
Signale und Slots	18
Objektorientierte Programmierung.....	19
QtDesigner	20
Ausblick.....	21
Anhang	23
Schlusswort	23
Programmcode.....	23

Quellen	24
Hilfsmittel und Bibliotheken	25

Einleitung

Es führt kein Weg daran vorbei: Vokabeln lernen gehört zum Alltag jeder Schülerin und jedes Schülers. Niemand macht es gerne, aber man muss durch. Daher stellt sich die Frage, wie sich Wortschatz mit möglichst wenig Zeitaufwand effizient und nachhaltig erweitern lässt.

Früher haben wir vermutlich alle 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 besteht darin, dass es für fast jedes Lehrmittel bereits Lernsets von Wörtern gibt, welche von anderen Benutzern zur Verfügung gestellt werden. Auch sonst muss nur eine Person in einer Gruppe 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 der Durchschnittsschüler 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 drei 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», wie ich es nannte, 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 der Programmiersprache 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 fünften 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.

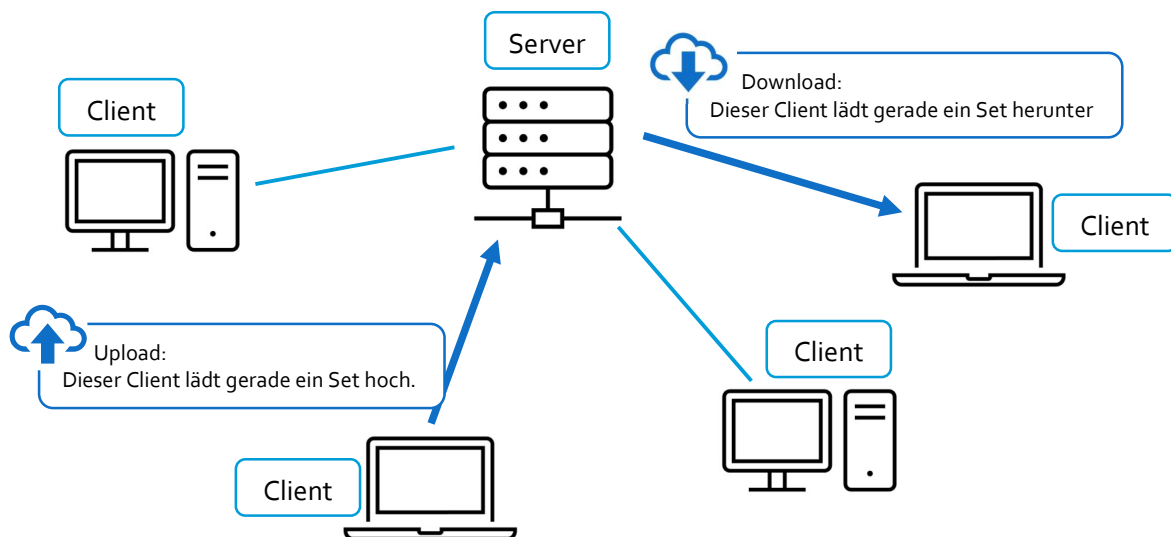
- 1) Mir ist es wichtig, dass es mit dem neuen Vokabeltrainer möglich ist, ganz unkompliziert selbst erstellte Lernsets mit anderen zu teilen, indem sie auf einen öffentlichen 'Marketplace' hochgeladen werden. Jeder andere Benutzer kann diesen durchsuchen und sich Lernsets herunterladen.
- 2) Da Anfangs erst wenige Lernsets zu Verfügung stehen werden, ist es mir wichtig, dass die Lernsets, welche zum Beispiel in Form von CSV-Dateien (Viele Programme bieten einen Export in dieses Format an) schon vorhanden sind, einfach importieren kann. Optimal wäre ein Tool, dass Lernsets direkt von Quizlet herunterlädt.
- 3) Die wichtigste Eigenschaft eines Vokabeltrainers ist es, ein effizientes Lernen zu ermöglichen. Gerade weil viele in der Praxis nicht regelmässig lernen, sondern erst die paar Tage vor der Prüfung, ist es von Bedeutung, 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, 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 für ein typisches Computerprogramm entschieden, welches offline nutzbar ist.

Die Zielgruppe sind Benutzer, die sich nicht mit den üblichen Lösungen zufriedengeben, 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 Benutzer (Client) angefordert werden können. Es ist ein entscheidendes Merkmal des Servers, dass dieser immer läuft.



Dieses Konzept benutze ich, um – wie in den Anforderungen beschrieben – die Möglichkeit zu erstellen, dass jeder seine Sets mit anderen teilen kann. Die Clients sind dabei die Benutzer beziehungsweise ihr 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 es 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 werden sowohl Daten auf dem Client wie auch Daten auf dem Server gespeichert. Beim Benutzer auf dem PC sind die ganzen Vokabeln und Sets gespeichert, sowie die Ordner Struktur (welches Lernset

in welchem Ordner ist). Der Server muss sich zusätzlich die verschiedenen Benutzer und ihre Passwörter merken.

Um all diese Informationen zu speichern, gibt es verschiedene Möglichkeiten. Ich könnte die Daten mit einer eigenen Struktur in einer einfachen Textdatei speichern. Alternativen dazu wären Formate wie JSON¹-Dateien und XML²-Dateien, welche bereits eine definierte Struktur haben.

Ich habe mich für eine Datenbank entschieden. In einer Datenbank werden Daten strukturiert gespeichert und sie bieten noch weitere Vorteile:

1. Die fixe Struktur von Datenbanken erlaubt es nicht, dass in der Datenbank Unordnung herrscht (Konsistenz und Integrität)
2. Eine Datenbank lässt sich einfach sortieren und durchsuchen. Dazu gibt es eine eigene Abfragesprache: SQL³.
3. Datenbanken sind für grosse Datenmengen und viele Benutzer optimiert und verbrauchen deshalb weniger Ressourcen.
4. In Datenbanken können Beziehungen zwischen den Daten gespeichert werden. Damit wird Redundanz vermieden. Der Fachbegriff dazu ist «relationale» Datenbank.

Stark vereinfacht kann man sich Datenbanken wie eine grosses Excel-Dokument vorstellen. Wie in einer Excel Arbeitsmappe, gibt es bei einer Datenbank unterschiedlicher Tabellen, in welchen verschiedenartige Informationen gespeichert werden. In der Datenbank des Clients gibt es verschiedene Tabellen. Sie speichern die heruntergeladenen Vokabeln (Tabelle «karte») und wie diese in Sets und Ordnern organisiert sind (Tabellen «vociset» und «ordner»).



Karte_ID	Wort	Fremdwort	...	Lernfortschritt	Set_ID
1	Apfel	apple	...	3	1
2	Pfirsich	peach	...	2	1
3	blau	bleu	...	1	2

Set_ID	Name	Sprache	...	Ordner_ID
1	Fruits	Englisch	...	1
2	Couleurs	Französisch	...	2

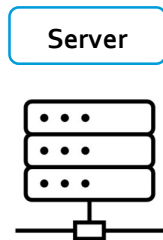
Ordner_ID	Name	...	Urordner_ID
1	Unit 3	...	0
2	Unité 1	...	0

¹ JavaScript Object Notation: Ein menschenlesbares Datenformat

² Extensible Markup Language: Ein weiteres verbreitetes Datenformat

³ Structured Query Language

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



Karte_ID	Wort	Fremdwort	...	Set_ID
1	Apfel	apple	...	1
2	Pfirsich	peach	...	1
3	blau	bleu	...	2

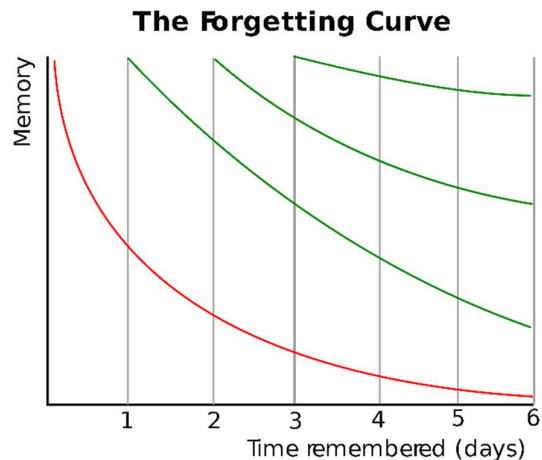
User_ID	Benutzername	E-Mail	...	Passwort-Hash
1	KSZler	meie.karl.2020@ksz...	...	f8g84ogaoafe...
2	Schüler16	martin436@gmail.com	...	14f766749559...

Set_ID	Name	Sprache	...	Downloads	User_ID
1	Fruits	Englisch	...	56	
2	Couleurs	Französisch	...	32	

Wie lerne ich am besten

Wiederholung macht den Meister

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 viel abgebildete Vergessenskurve. Die rote Linie zeigt die Abnahme des verbleibenden Wissens über die Zeit. Wenn jedoch nach einem Tag die Lerninhalte nochmals wiederholt werden, geraten sie nicht mehr so schnell in Vergessenheit. (1. Grüne Kurve) Mit jeder Wiederholung wird die Kurve flacher. Nach mehrmaligem Wiederholen sinkt allerdings der Verbesserungseffekt (Die Kurve flacht kaum mehr ab). Überlernen lohnt sich folglich nicht, da es nur kleine Vorteile bringt.



Die magische Zahl 7 ± 2

In diesem Abschnitt geht es um die von George Miller beschriebene 'magische Zahl'. Ich habe einen Selbstversuch vorbereitet, mit dem Sie versuchen können, ihre eigene 'magische Zahl' zu finden. Auf den folgenden Seiten befinden orange eingefärbte Kästen, in denen der Selbstversuch beschrieben ist. Achtung: Vorblättern verfälscht das Ergebnis!

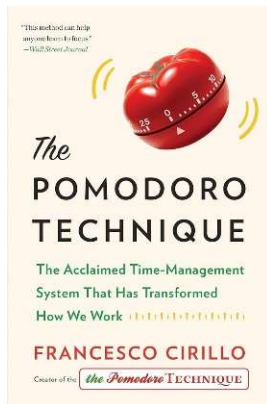
Selbstversuch

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.

Der 1956 von George A. Miller veröffentlichte Artikel «The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information» gehört zu den meistzitierten Artikeln der Psychologie. Darin beschreibt Miller den Umstand, dass wir uns nur ungefähr 7 Informationseinheiten im Kurzzeitgedächtnis merken können.

Diese Erkenntnis hat konkrete Auswirkungen darauf, wie wir am effektivsten neue Wörter lernen. Ein Wort zu erlernen ist ein Prozess. Beim ersten Durchlesen, versucht das Gehirn sich die Bedeutung einzuprägen, beim zweiten oder dritten Mal versucht es bereits die richtige Schreibweise zu lernen. Während diesem Prozess sollte das Wort im Kurzzeitgedächtnis bleiben. Miller's Law angewendet heisst darum, dass nur immer sieben 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. Ablenkungen sind überall und viele leiden am Drang, auch nur kurz nach neuen Nachrichten auf WhatsApp zu checken. Dem Italiener Francesco Cirillo ging es ähnlich. 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 weiterentwickelt und ein Buch darüber geschrieben (siehe Bild).

Die Technik funktioniert folgendermassen:

1. Sich eine Aufgabe stellen
2. Eine Uhr auf 25 Minuten einstellen
3. 25 Minuten fokussiert durcharbeiten
4. Alles stehen lassen und 5 Minuten Pause machen
5. Das lässt sich mehrmals wiederholen. Nach vier 'Pomodori' sollte eine grössere Pause von bis zu 30 Minuten eingelegt werden.

Die Idee ist, dass mit einem klar definierten Ziel und beschränkter Zeit die Konzentration erhöht und die Versuchung der Ablenkung kleiner ist.

Selbstversuch

Haus

Quadrat

Tür

Buch

Apfel

Auto

Schuh

Kugel

Band

Schlüssel

Fisch

Box

Kopf

Hammer

Pfeil

Katze

Blume

Milch

Baum

König

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.

Selbstversuch

Schritt 2: Schreiben Sie nun ohne nachzuschauen alle Wörter, an die sie sich erinnern können, auf ein Blatt Papier. Sie haben dafür etwa eine halbe Minute Zeit.

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, leite ich daraus einiges für das Vokabellernen ab.

1. Da das handschriftliche Schreiben länger dauert, bleibt dem 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, mit dem Blick mehrmals zwischen dem Wort und dem selbst geschriebenen zu wechseln. Dies verbessert den Lerneffekt weiter.

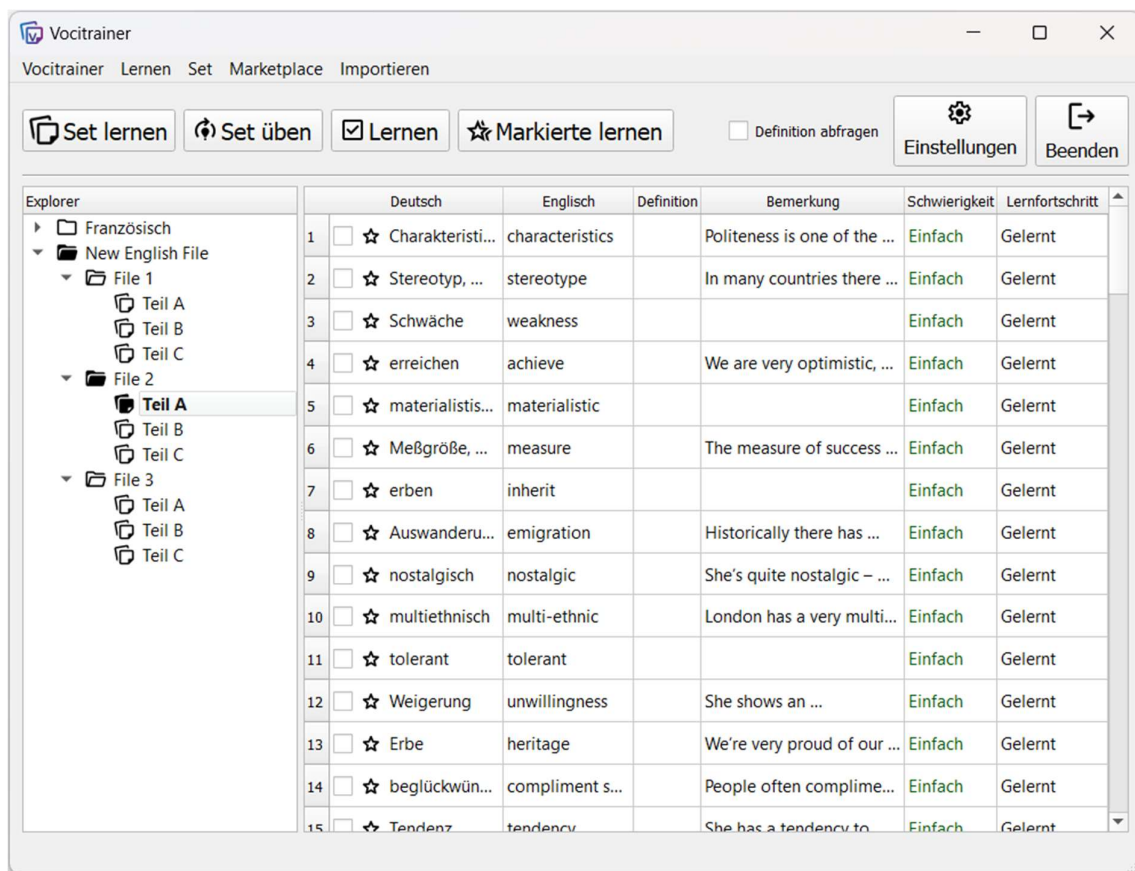
Die Frage ist, wie sich diese Tatsache in einen digitalen Vokabel-Trainer einbinden lässt. Mit einem digitalen Programm lassen sich schriftliche Eingaben nur schwer verwerten. Deshalb habe ich den Weg gewählt, dass der Vokabeltrainer immer, wenn ein Wort zum ersten Mal erscheint, eine eigene Meldung kommt. Es empfiehlt sich, dann das Wort einmal handschriftlich abzuschreiben.

Selbstversuch

Schritt 3: Zähle, wie viele Wörter sie richtig haben. Vermutlich konnten sie sich an 5-9 Wörter erinnern. Diese Limitierung ist es, welche «Millers Law» beschreibt.

Funktionen

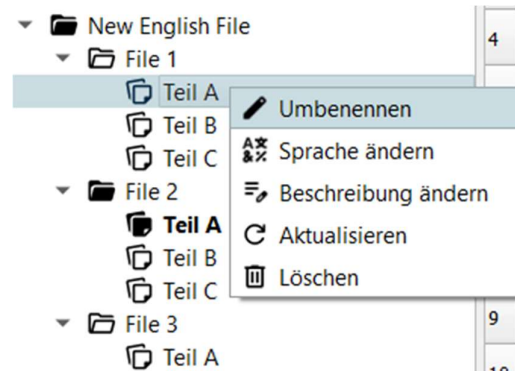
In diesem Kapitel möchte ich gerne genauer auf die Funktionen meines Vokabeltrainers eingehen. Besonders als Leser, der den 'Vocitrainer' nicht selbst ausprobiert hat, bekommen sie hier einen Einblick.



So sieht der Vokabeltrainer aus, nachdem er gestartet wurde.

Das Hauptfenster

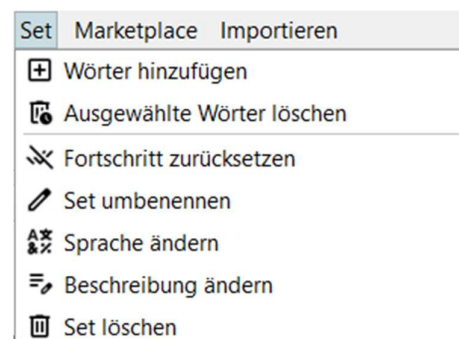
Das Hauptfenster besteht aus mehreren Teilen. Links im Fenster ist der Explorer positioniert. Er zeigt die hierarchische Organisation der Sets durch Ordner. Öffnet man ein Set mit einem Doppelklick, wird die ganze Struktur bis dahin fett eingefärbt, um die Übersicht zu erhöhen. Da eine intuitive Bedienung hier sehr wichtig ist, habe ich viel Zeit investiert um alle Drag-and-drop Aktionen zu programmieren. Zusätzlich sind über ein Kontextmenü alle Aktionen verfügbar, um die Elemente im Explorer zu bearbeiten.



Auf der rechten Seite befindet sich die Übersicht des Sets. In einer Tabelle werden hier alle Karten aufgelistet, welche das ausgewählte Set enthält. In einer eigenen Spalte ist gleich farblich hervorgehoben die vom intelligenten Lernmodus ausgemachte Schwierigkeit des Wortes sichtbar. Zusätzlich lassen sich die Wörter gleich in der Tabelle selbst bearbeiten. Der dafür nötige Doppelklick in das Feld verhindert ungewollte Änderungen, wie man sie vielleicht von Excel kennt.

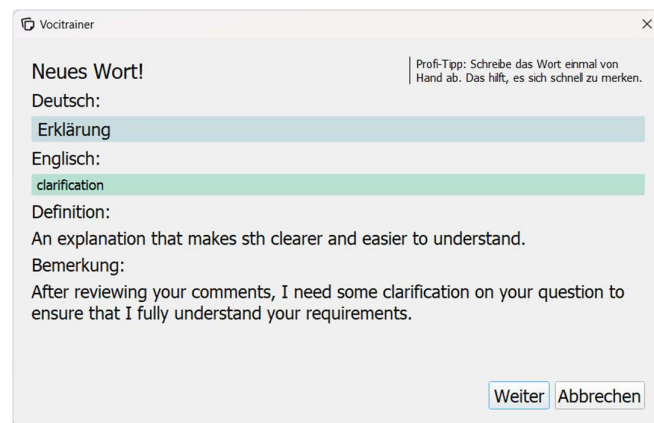


Am oberen Rand des Fensters befinden sich die Menüs. In den Menüs sind alle Aktionen organisiert, welche der Vocitrainer bietet. Aus dem Lernen-Menü lassen sich alle Lernmodi starten. Im Set-Menü sind alle Aktionen zum Bearbeiten des offenen Sets zusammengefasst. Von «Marketplace» aus lassen sich Sets mit anderen Teilen und herunterladen. «Importieren» bietet momentan erst eine Möglichkeit, Sets von extern zu importieren. Allerdings ist CSV⁴ hierbei das verbreitetste Format.



⁴ CSV: Comma Separated Values

Das Training



Für das Training öffnet sich ein eigenes Fenster. Dieses hat drei Zustände. Wann welcher Zustand aufgerufen wird bestimmt der Lernmodus. Derer gibt drei. Der Lernmodus «Set üben» ist noch nicht programmiert.

Der intelligente Lernmodus «Set lernen» beachtet die im Kapitel «Wie lerne ich am Besten» beschriebenen Kenntnisse, um einen schnelleren Lernerfolg zu erreichen. Bevor der Vokabeltrainer ein Wort zum ersten Mal abfragt, wird es dem Benutzer gezeigt, sodass er die Möglichkeit hat, es sich einzuprägen, optimalerweise indem er es einmal handschriftlich abschreibt. Danach werden die Wörter abgefragt. Vokabeln, die der Lernende mehrmals falsch beantwortet, werden als schwieriger markiert (siehe Text im Fenster links unten) und mehrmals abgefragt. Wie tolerant der Vokabeltrainer sein soll bzw. wie schnell er ein Wort als schwieriger markiert lässt sich einstellen. Dabei werden

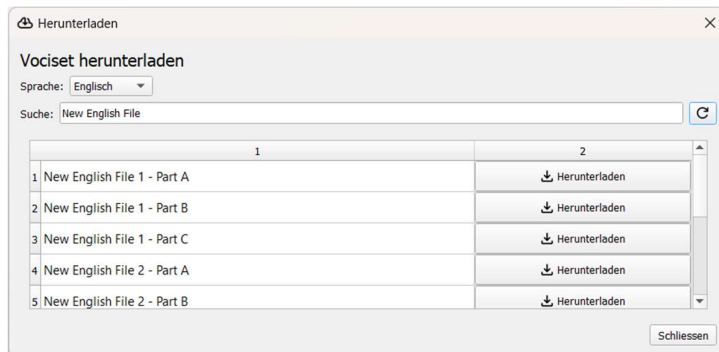
jeweils 7 (± 2 personalisierbar) Wörter auf einmal gelernt um das Kurzzeitgedächtnis nicht zu überlasten.

In den Lernmodi «Lernen» und «Markierte lernen» werden alle Wörter einfach abgefragt, bis sie einmal richtig beantwortet wurden.

Zusätzlich kann mit dem Setzen eines Häkchens eingestellt werden, dass der Vokabeltrainer das Wort mit seiner Definition abfragt.

Der Marketplace

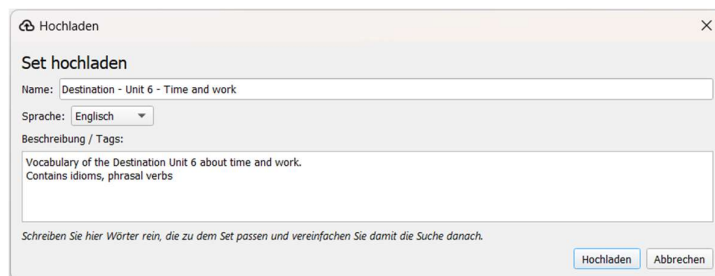
Mir war es wichtig, dass es ganz einfach ist, seine erstellten Sets mit anderen zu teilen und selbst Sets von anderen herunterzuladen. Die Umsetzung davon ist der «Marketplace». Der Marketplace ist eine Plattform zu der jeder mit dem Vocitrainer-Programm Zugang hat. Der Benutzer erstellt sich dort ein Konto und schon kann er auch selber Sets hochladen. Das Herunterladen von Sets ist sogar ganz ohne Anmeldung möglich.



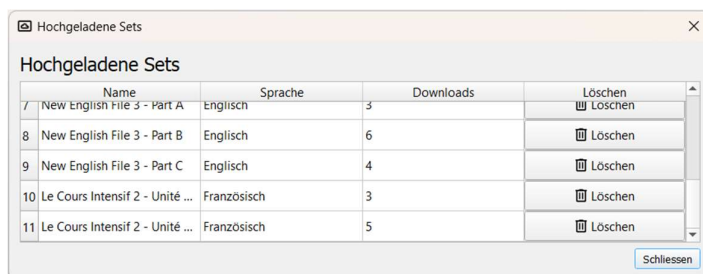
Die Sets lassen sich einfach durchsuchen und mit einem Klick herunterladen.

Alle Funktionen zum Marketplace sind in dem gleichnamigen Menü zusammengefasst.

Die Bedienung ist sehr intuitiv. Deshalb kommen hier nur noch ein paar Bilder für die Leser, welche den Vokabeltrainer nicht selbst bereits ausprobiert haben.



Beim Hochladen eines Sets wird ein Name, eine Sprache und eine Beschreibung dazugeschrieben.



In der Übersicht der hochgeladenen Sets können diese wieder gelöscht werden.

Qt-Framework

Das Qt-Framework (Qt ausgesprochen wie das Englische 'cute') ist wohl die wichtigste Komponente meiner Arbeit. Ich habe viel Zeit gebraucht, um dieses umfangreiche Framework auch nur im Ansatz kennen zu lernen. Deshalb möchte ich ihm ein eigenes Kapitel widmen und erklären, wie Qt im Grundsatz funktioniert. Es gibt ein Tutorial, wie man als Anfänger ein kleines Programm schreiben kann.



Dieses Kapitel ist sehr technisch geschrieben und besonders für die Leser geeignet, die Python bereits kennen und interessiert sind, wie grafischen Benutzeroberflächen funktionieren.

Was ist ein Framework

Framework nennt man eine grosse Sammlung von bereits geschriebenen Programmen. Diese kümmern sich zum Beispiel darum, wie ein Fenster genau auf dem Bildschirm kommt und wie dabei die Kommunikation mit dem Betriebssystem abläuft. Frameworks sind essenziell, weil sie einem viel Arbeit abnehmen. Es ist nicht möglich, für jedes kleine Projekt alles von Grund auf zu programmieren.

Im Unterschied zu einer Bibliothek, die eine ähnliche Funktion hat, bestimmt ein Framework die Struktur des Programmes und setzt eine definierte Anwendung voraus.

Funktion

Qt ist ein Framework für grafische Benutzeroberflächen auch GUI (Graphical User Interface) genannt. Das Framework ist sehr verbreitet, da es einen grossen Funktionsumfang hat, Open Source ist (jeder kann es verwenden) und in vielen Programmiersprachen verfügbar ist (siehe Absatz [PyQt](#)).

Qt wird unter anderem vom legendären VLC Media Player verwendet und weiteren Programmen wie: Adobe Photoshop Elements, Google Earth, VirtualBox, Spotify (Linux Version) oder dem Simulationsprogramm der ESA.⁵ Alles, was bei meinem Vokabeltrainer zu sehen ist, wurde mit Qt umgesetzt.

⁵ Siehe die Quelle: Qt Bibliothek (Wikipedia)

PyQt

Qt ist, wie viele Bibliotheken für Python in C++ geschrieben, da diese Programmiersprache viel schneller und «genauer» ist und sich deshalb besser dafür eignet. Das heisst gleichzeitig, dass ich Qt in Python nicht einfach so verwenden kann. Es braucht ein Glied zwischen Python und dem in C++ programmierten Qt, welches die beiden verbindet. Das nennt man einen Wrapper. Für Qt gibt es 2 Python-Wrapper. PyQt und PySide. Es gibt keine grossen Unterschiede zwischen den beiden; ich habe mich für PyQt entschieden. PyQt implementiert über 1000 Klassen von Qt.⁶

Beispiel eines Miniprogrammes

Das ist ein Beispiel eines minimalistischen PyQt-Fensters.

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QPushButton

# Erstellen einer Qt-Anwendung [1]
app = QApplication(sys.argv)

# Erstellen eines Fensters [2]
window = QWidget()
window.setWindowTitle('Ein PyQt-Fenster')
window.setGeometry(100, 100, 300, 100) # Position und Größe des Fensters

# Erstellen eines Labels mit Text "Hallo Welt" [3]
label = QLabel('Hallo Welt', window)

# Position des Labels im Fenster festlegen
label.move(100, 20)

# Fenster anzeigen
window.show()

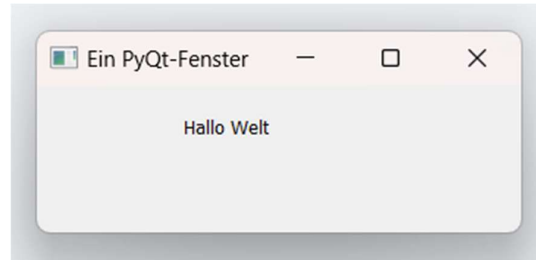
# Anwendung ausführen [4]
sys.exit(app.exec_())
```

Zuerst erstellen wir eine Qt-Anwendung [1]. Dann wird das eigentliche Fenster initialisiert [2]. Dabei kann jedes Widget ein Fenster sein. Es gibt jedoch extra Klassen dafür wie QDialog oder

⁶ Siehe Quelle: PyQt5 Dokumentation

QMainWindow. Nun fügen wir dem Fenster ein einfaches Label hinzu [3]. Als letztes starten wir das Programm [4]. Ab dann läuft die Event Loop, bis das Programm geschlossen wird.

Wenn wir das Programm ausführen, erscheint diese Fenster:



Signale und Slots

Eine wichtige Frage bei grafischen Anwendungen ist immer, wie werden Benutzereingaben verarbeitet? Qt hat hier sein eigenes Konzept namens Signals & Slots.

Elemente können in Qt bei einer Aktion ein Signal senden. Zum Beispiel ein Button der geklickt wird. Diese Signale können mit einem Slot verbunden werden. Dieser Slot ist meistens eine selbst geschriebene Funktion, welche die Eingabe verarbeitet.

In unserem Beispiel von oben könnten wir Folgendes ergänzen.

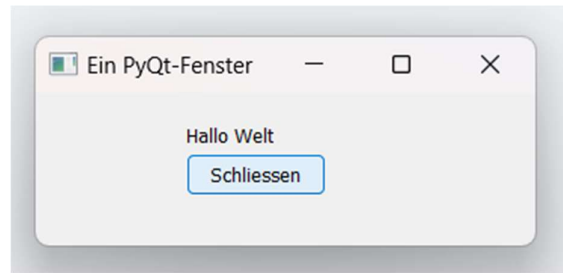
```
# Slot für das Schliessen Signal des Buttons [1]
1 usage
def fenster_schliessen():
    window.close()

# Einen Schliessen Button einfügen
cmd_schliessen = QPushButton("Schliessen", window)
cmd_schliessen.move(100, 40)
cmd_schliessen.clicked.connect(fenster_schliessen) # [2]
```

Wir erstellen eine Funktion, die das Fenster schliesst [1] und verbinden diese als Slot mit dem 'clicked'-Signal des Buttons, den wir erstellt haben [2].

Sobald man nun auf diesen neuen Button klickt, wird das Fenster geschlossen.

Wer das Beispiel, das wir bis jetzt betrachtet haben, weiter in Gedanken zu einem grösseren Projekt ausweitet, merkt, dass das schnell unübersichtlich werden kann. Deshalb brauchen wir zwei Hilfsmittel.



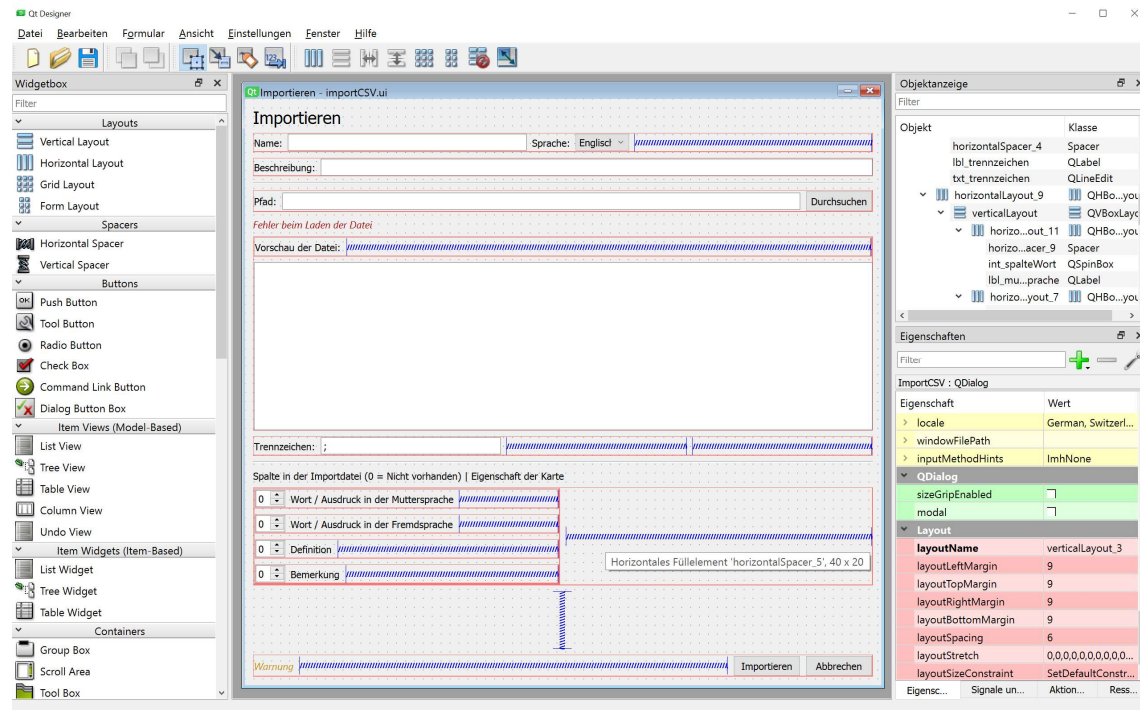
Objektorientierte Programmierung

Objektorientierte Programmierung kurz OOP wurde mit dem Aufkommen von grafischen Benutzeroberflächen erfunden. Bei den Unmengen an Objekten, Variablen und Funktionen braucht es eine Struktur, welche das Ganze organisiert. OOP heisst kurz gesagt, dass es Objekte, zum Beispiel Klassen gibt, denen Funktionen und Variablen zuordnet werden, welche Methoden und Attribute heissen. So hat beispielsweise jedes QLabel die Methode `QLabel.setText()`. Fenster lassen sich am besten in selbst geschriebenen Klassen zusammenfassen.

QtDesigner

Das zweite Hilfsmittel erleichtert das Leben sehr. In unserem Beispiel zuvor haben wir das Label und den Button selber erstellt und platziert. Das ist bei komplexeren Fenstern kaum mehr möglich. Man müsste immer ausprobieren wie das Fenster aussieht und dann wieder verbessern.

Die Lösung ist der mit Qt mitgelieferte QtDesigner. Er ermöglicht es, das Fenster per Drag and Drop zusammenzubauen.



Hier ein Bild des Fensters zum Importieren aus CSV-Dateien. Der wichtigste Bereich ist das Fenster in der Mitte. Links sind die verfügbaren Widgets und rechts unten der Bereich um die Eigenschaften des angewählten Widgets zu bearbeiten.

Falls Sie selbst gerne mit Python programmieren und interessiert sind, kann ich die Webseite www.pythonguis.com für den Einstieg sehr empfehlen.

Ausblick

Der Vokabeltrainer als Produkt meiner Maturaarbeit ist so weit abgeschlossen. Ich möchte aber nicht, dass die ganze Arbeit wie eine ungültig gewordene Telefonnummer in Vergessenheit gerät und nie mehr benutzt wird, im Gegenteil! Während des Programmierens habe ich immer besonders auf Erweiterbarkeit geachtet. Das beginnt in den kleinen Details und geht so weit, dass es bereits ein Button gibt für den Lernmodus «Set üben» der noch gar nicht implementiert ist, wobei es auch nie das Ziel war, dass noch vor Abgabe zu erreichen. Das Programm ist erst das Grundgerüst für viel mehr. Und der modulare Aufbau ermöglicht es mir auch, zum Beispiel ganz einfach neue Lernmodi zu implementieren. Es wäre sogar möglich ein Plugin-System dafür zu erstellen.

In der letzten Phase des Projekts habe ich mir viele Gedanken darüber gemacht, in welche Richtung sich der Vocitrainer weiterentwickeln soll. Es gibt viele Möglichkeiten.

Eine Möglichkeit wäre, den Vocitrainer auf neue Plattformen zu bringen: Apps sind überall und jedes Unternehmen hat seine App. Ich könnte eine mobile Version des Vocitrainers für iOS und Android programmieren, die ich dann in den jeweiligen App Stores zum Download anbiete. Allerdings glaube ich nicht, dass das Vokabellernen auf mobilen Geräten eine Zukunft hat. Zu Nahe sind die Ablenkung und generell ist das Smartphone ein Konsum- und nicht ein Arbeitsgerät. Ausserdem fehlt eine Tastatur oder andere Möglichkeiten, um Wörter einzugeben und somit aktiv zu lernen. Unser Gehirn lernt aber nur, wenn es selbst aktiv wird.

Eine andere Variante wäre, das Design zu überarbeiten und das Programm durch kleine Spiele und der Funktion von Lernserien und Lernabzeichen spannender zu machen. Auch hier glaube ich jedoch, dass dies nicht zielführend ist. Kleine Spiele zu programmieren, bedeutet nur, mit blinkenden Inhalten und Herausforderungen mit TikTok, Instagram und Videospielen in einen Wettlauf um die Aufmerksamkeit zu treten, das man ohnehin nicht gewinnen kann. Das Konzept von Lernserien und Abzeichen mag spannend klingen. Auf den zweiten Blick erkennt man aber, dass dies nicht viel bringt: Warum sollte ein Schüler in den Ferien Französisch lernen, um seine Lernserie aufrecht zu erhalten. Ein Update im Design tönt zwar nicht schlecht, bringt aber keine unmittelbaren Vorteile, ist aufwendig und langweilig.

Ich möchte den Vocitrainer in eine andere Richtung weiterentwickeln. So wie der Marketplace jetzt ist, hat er keine Zukunft. Es müssen Funktionen nachgerüstet werden, um Missbrauch zu verhindern. Dazu bräuchte es eine E-Mail-Überprüfung, Meldeverfahren und meine Zeit, mich mit alledem zu beschäftigen, vorausgesetzt, jemand verwendet den Vokabeltrainer überhaupt.

Ich glaube die Lösung dieses Problem liegt darin, den Vocitrainer und besonders den Serverteil davon Groupware tauglich zu machen. So nennt man Software, welche spezielle Funktionen für den Gebrauch in Organisationen wie einer Schule bereitstellt.

Meine Vision ist dabei folgende: Ich möchte mit den Sprach-Fachschaften der Schule ins Gespräch kommen. Ziel ist es, ihnen die Nachteile von Quizlet und anderen je nach Lehrer individuellen Lösungen aufzuzeigen. Der Vocitrainer könnte hier Lösung bieten. Die Schule hat dann einen eigenen Server, welchen die Schüler auswählen können. Auf diesem Server kann man sich nicht selbst registrieren. Nur die Lehrer haben ein Konto und damit können auch nur sie Sets hochladen.

Ich habe die Anpassungen, die dafür nötig wären, zwar schon geplant und werde sie vermutlich auch bald umsetzen. Trotzdem ist dies erst eine Vision und der Weg zur Umsetzung ist ein Abenteuer, wie es der Weg bis zum heutigen Stand war.

Anhang

Schlusswort

Vor einem Jahr nahm in mir die Idee Gestalt an, einen Vokabeltrainer zu programmieren. Ich wusste damals kaum etwas über die Programmierung komplexer grafischer Benutzeroberflächen mit Python. Seitdem hat sich vieles getan. In der sich rasch entwickelnden Welt der Informatik ist Lernbereitschaft eine wichtige Eigenschaft. Ich habe viel gelernt und es hat mir viel Spass gemacht. Es ist eine grosse Freude, wenn das neu Gelernte gleich einzusetzen kann und zu sehen, dass es funktioniert. Der Vocitrainer besteht inzwischen aus über 6'000 Zeilen Code, die alle nötigen Funktionen liefern. Ich hoffe, dass er noch dem einen oder anderen Lernenden eine Hilfe sein wird.

Ich möchte mich bei allen bedanken, die mich auf meinem Abenteuer, denn das war es, begleitet haben. Ein besonderer Dank geht an Herrn Kubba, der mich betreut hat. Er hat mir immer geduldig zugehört, während ich mit Begeisterung meine Fortschritte präsentierte. Aber auch in den Situationen, als ich irgendwo festhing, war Verlass auf Herrn Kubba. Mit seiner Erfahrung konnte er mir immer die entscheidenden Tipps geben.

Ich danke Mirco Triner. In vielen Gesprächen mit ihm konnte ich meine Ideen reflektieren und weiterentwickeln.

Besonderer Dank geht auch an meine Eltern und Geschwister. Mit viel Geduld haben sie mir zugehört und viel Verständnis aufgebracht, wenn ich am Abend manchmal in mein Zimmer verschwand und für nichts mehr zu haben war: «Ruhe, Maturaarbeit!»

Ein letzter Dank auch an den Leser, der sich selbst noch für das Schlusswort bemüht. Ich hoffe, die Lektüre hat sich gelohnt.

Fragen zum Vocitrainer sind stets willkommen.

Ich bin per Mail erreichbar: samuel@barmet.ch

Programmcode

Der Programmcode ist öffentlich zugänglich auf GitHub. Suchen sie nach «Vocitrainer» oder öffnen sie folgenden Link im Browser:

<https://github.com/samuelx12/Vocitrainer>

Quellen

3 : 0 – Stift siegt gegen Tastatur (Dr. Martin Tomasik)

<https://www.psychologie.uzh.ch/de/bereiche/dev/lifespan/erleben/berichte/handschrift.html>
(5.12.2023)

Die Gedächtnismodelle (Anna Westhoff)

<https://www.brain-effect.com/magazin/dr-karsten-das-kurzzeitgedaechtnis> (14.06.2010)

Halten Sie sich kurz! Mit kurzen Lerneinheiten zum Erfolg (Katharina)

<https://blog.brain-friendly.de/2013/04/halten-sie-sich-kurz-mit-kurzen-lerneinheiten-zum-erfolg/> (19.04.2013)

Lernpausen sind gut fürs Gedächtnis (Max-Planck-Gesellschaft)

<https://www.mpg.de/17299970/0728-psy-lernpausen-sind-gut-fuers-gedaechtnis-155111-x>
(28.07.2021)

Miller's Law: Die wichtigste Regel im UX-Design (Jessy Kösterke)

<https://t3n.de/news/millers-law-wichtigste-regel-ux-1011492/> (14.04.2018)

Pomodoro-Technik: Wie Sie Ablenkungen widerstehen und Ihre Produktivität erhöhen (IONOS)

<https://www.ionos.de/startupguide/produktivitaet/pomodoro-technik/> (22.05.2023)

Produktiver Arbeiten mit der Pomodoro-Technik (Juliane Felder)

<https://www.fhnw.ch/plattformen/zentrumschreiben/2022/02/18/produktiver-arbeiten-mit-der-pomodoro-technik/> (05.12.2023)

Python 3 – Das umfassende Handbuch (Johannes Ernesti, Peter Kaiser)

Rheinwerk Verlag, Bonn, 7., aktualisierte Auflage 2023

PyQt5 Dokumentation (Riverbank Computing)

<https://www.riverbankcomputing.com/static/Docs/PyQt5/introduction.html> (15.01.2024)

PyQt5 Tutorial – Getting started with PyQt5 (Martin Fitzpatrick):

<https://www.pythonguis.com/tutorials/creating-your-first-pyqt-window/> (13.09.2023)

PyQt5 Python 3 Tutorial (TechWithTim)

<https://www.youtube.com/playlist?list=PLzMbGfZo4-IB8MZfHPLTEHO9zJDDLpYj> (24.4.2020)

Qt Bibliothek (Wikipedia)

[https://de.wikipedia.org/wiki/Qt_\(Bibliothek\)](https://de.wikipedia.org/wiki/Qt_(Bibliothek)) (23.09.2023)

Spaced Repetition (Wikipedia)

https://de.wikipedia.org/wiki/Spaced_repetition (05.12.2023)

The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information (George A. Miller) (März 1956)

Verfügbar unter:

<https://labs.la.utexas.edu/gilden/files/2016/04/MagicNumberSeven-Miller1956.pdf>

(Bild) The Pomodore Technique Book Cover

<https://www.orellfuessli.ch/shop/home/artikeldetails/A1043641537> (26.12.2023)

Vergessenskurve (Wikipedia)

<https://de.wikipedia.org/wiki/Vergessenskurve> (18.02.2022)

Vokabeln lernen im digitalen Zeitalter (Claudio Abart, Joël Mathys)

Eine Maturaarbeit aus dem Jahr 2015/16 verfügbar im Info-Z der Kantonsschule Zug.

Hilfsmittel und Bibliotheken

DB Browser (SQLite): <https://sqlitebrowser.org/>

GitHub: <https://github.com/>

Google Fonts: <https://fonts.google.com/>

PyCharm: <https://www.jetbrains.com/de-de/pycharm/>

Python: <https://www.python.org/>

Pythons Standardbibliothek: Socket, pickle, ssl, sqlite, csv, typing, dataclasses

PyQt5: <https://pypi.org/project/PyQt5/>, <https://www.riverbankcomputing.com/software/pyqt/>

Rich: <https://rich.readthedocs.io/en/stable/>