

Richtlinie zum „Software-Sprint“

BLN ABC – (Berlin ABC) Eine mobile Webapp für die Grundalphabetisierung

Schlussbericht

Zuwendungsempfänger:

Jens Kreitmeyer

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS24S33** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen?

BLN ABC ist konzipiert als ein "Primer" ("Vorkurs A0") für die Entwicklung von ersten, elementaren Schreib-Lese-Fertigkeiten, die auf dem Anfangsniveau von Alphabetisierungskursen auf GER Level A 1.1 (Deutsch als Zweitsprache) liegen.

An erster Stelle stehen dabei kognitive Automatisierung der Phonem-Graphem-Korrespondenz (Laut-Zeichen-Entsprechung), Lautdifferenzierung/Artikulation, phonologische Bewusstheit, Entwicklung der "Motorprogramme" für das (handschriftliche) Schreiben.

Mit der App sollen vor allem auf dieser Stufe oft vorhandene (und in der eigenen Unterrichtspraxis beobachtete) "Lücken" geschlossen werden. „Rückstand“ kann aufgeholt werden, wenn Übungen nach Lerntempo/Lernstand angepasst mitgegeben werden können und Lernende, über die Möglichkeiten im Kurs hinaus, damit individuell unterstützt werden.

„Lesen-Schreiben-Lernen“ ist, besonders am Anfang, ein mühsamer Vorgang, der hauptsächlich auf Wiederholungen und Wiederholungen und Wiederholungen basiert. Alles, was dies etwas leichter, unterhaltsamer und (vielleicht) effektiver machen könnte, wurde in Betracht gezogen.

Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

Die Meilensteine sahen die Überarbeitung des Moduls „Nachspuren“ (und der dafür erforderlichen Grafik-Pipeline) vor, die Ausarbeitung des „Meta“-Moduls „RecorderPlayer“ (overlay der

phonetischen Übungen über alle visuellen/motorischen Teile), Weiterentwicklung des Moduls „Verschriften“ (alias Silbenschieber/Stepper, s. Punkt Text-to-Speech) und Einbettung aller Bestandteile in ein übergreifendes GUI-Framework, ursprünglich mit „Swiper“ (vergl. Punkt 5).

Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Welche Bezüge gibt es zu den Themenfeldern „Civic Tech“ und „Data Literacy“ des Software Sprints oder zu weiteren gesellschaftlich relevanten Zielen bzw. Lösungsansätzen?

Die App richtet sich an LernerInnen und Lehrkräfte vorrangig im Bereich Deutsch als Zweitsprache/erwachsene Zweitschriftlerner. Darüber hinaus kann die App im Prinzip in allen Alphabetisierungskontexten eingesetzt werden.

Literalität (Alphabetisierung) wird als „kognitive Schlüsseltechnologie“ aufgefasst, die eine Vorbedingung für den (unmittelbaren) Zugang zu Wissen/Information und für Teilnahme/ Teilhabe ist.

In Zeiten von „fake news“ und LLMs wird diese Kompetenz umso wichtiger, je mehr Zweifel an Authentizität und Autorität von Quellen aufkommen.

Die App soll begleitend zum Unterricht in Alphakursen eingesetzt werden können, aber ggf. auch autonomes (d.h. selbständiges und unbegleitetes) Lernen unterstützen können.

Literalität begünstigt Autonomie und Inklusion und ist als eine Teilmenge von „data literacy“ und „civic tech“ anzusehen.

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt?

„APP-CORE“:

LESEN (VORSCHRIFT), SCHREIBEN (NACHSPUREN), HÖREN, SPRECHEN

Die 4 „Kernmodule“ der App sind prototypisch (vollständig, robust und "serienreif“) umgesetzt worden.

LESEN („WATCH“) rendert die Animation der Schreibvorlage (Template). Dies geschieht verbindlich mit Strichrichtung und Strichreihenfolge, für alle Glyphen aus dem Font "Anfangsschrift" (lateinisch-deutsches Grundalphabet Aa-Zz, ÄäÖöÜü, ß) und alle phonetisch relevanten Buchstabengruppen („mehnteilige Basisgrapheme“, wie „ei/ie“, „sch“ usw.).

Wie die mehrteiligen Grapheme, können auch andere beliebige, frei definierbare Zeichenfolgen (wie z.B. „-ling“ oder „mpfst“) wysiwyg zusammengesetzt werden (vergl. LAUNCHER).

Die Vektordaten der Einzelglyphen, also der komplette Font, sind in einem kompakten JSON Format gespeichert.

Die Animation kann in verschiedenen „Flavors“ dargestellt werden, bisher als „Karaoke“-Punkt oder als „Kometenschweif“, die verschiedene Schwierigkeitsstufen darstellen (und den Neuigkeitswert verlängern).

SCHREIBEN („WRITE“) umfasst das Nachspuren aller Grapheme bei interaktivem, korrigierendem Feedback, d.h. Fehler in Bezug auf Richtung oder Reihenfolge und Grad der Abweichung von der

"Ideallinie". Die Generierung dieser "reaktiven Schreibspur", die frei skalierbar und in der Auflösung variabel ist, erfolgt durch Bezier-Interpolation und einen „crawler“-Algorithmus zur Laufzeit.

REC(ORDER)PLAY(ER) „REPLAY“ kombiniert das HÖREN der zu dem gezeigten Graphem gehörenden Lautierung (dem Phonem) mit dem SPRECHEN, d.h. der gerade gehörte Laut kann sofort nachgesprochen und aufgezeichnet („kopiert“) werden.

REPLAY gestattet das verzögerungsfreie Hin- und Herschalten zwischen Audiovorgabe und Kopie — und macht damit unmittelbares, iteratives Vergleichen und Verbessern möglich.

Das Modul ist zusätzlich zu allen anderen Bearbeitungsformaten aufrufbar (WATCH, WRITE, QUIZ, STEPPER).

EXTENDED FEATURES:

KONFIGURATION, PROGRESSION, LERNSTAND

Die erweiterten Features der App sind noch nicht fest in die App eingebunden. Das komplexeste Modul STEPPER („Silbenschieber“) ist aber als „working prototype“/proof-of-concept benutzbar und evaluierbar. Die Module LAUNCHER und PORTFOLIO sind eine erste Iteration von Konzept bzw. code für den weiteren Ausbau der App.

LAUNCHER

Das Modul LAUNCHER (Icon „Home“) besteht aus einer „sprechenden“ Tastatur und Menüs, das sind vorkonfigurierte Filter zum Aufruf verschiedener Zusammenstellungen von Graphemen als Übungen. Im GUI des Prototype wurde es zunächst durch eine einfache Auswahlliste ersetzt. (Das Modul ist extern aufrufbar (Homepage) .../alphasprint/LAUNCHER/main_chrome.html)

Diese Auswahlliste wird aus einer frei editierbaren Sammlung („COLLECTION“) erzeugt, die als normaler String/Text, wysiwyg wie „S, s, sch, Sch“, in der HTML Index-Datei eingegeben/ausgelesen werden kann. (Lediglich Glyphenvarianten benötigen eine zusätzliche Indexnummer, z.B. „g“, „g2“). Die Liste behält ihre Reihenfolge, d.h. wird nicht alphanumerisch sortiert, so dass feste Auswahlen angelegt werden können. Darüber hinaus können beliebige Zeichenfolgen (strings), in der Länge im Prinzip nur begrenzt durch die Bildschirmgröße, eingegeben werden, z.B. für Affixe oder morphologische Einheiten (wie „-ig“, „-lich“, „-ung“ usw. usf.) oder auch kurze Wörter.

COLLECTIONS sind in dieser Form bereits die programmatische Schnittstelle zum (späteren) LAUNCHER, der Lehrkräften als ein einfaches Interface für die Zusammenstellung eigener individualisierter Übungen dienen soll. Damit kann auch die Synchronisierung mit den Kanons der gängigsten Alpha-Lehrwerke erfolgen.

STEPPER

Der STEPPER alias „Silbenschieber“ (Home) .../alphasprint/SILBENSCHIEBER/audioautocomp.html ist m.E. das innovativste und fortschrittlichste Feature der App, konnte aber im Förderzeitraum, aufgrund der Komplexität des zugrundeliegenden Softwarestacks, nur als Demo weiter verbessert und vervollständigt werden.

Dazu wurden 8.500 Audioclips "gesichtet" (abgehört) und ca. 800 phonetische „Schnipsel“ neu eingesprochen und geschnitten, um das vollständige Lautinventar (isolierte, akkumulierend zusammengesetzte Wortfragmente) bereitzustellen. Die neu vertonten Fragmente belegen die

ersten 1 bis 3 Stellen in Wörtern und können in den entsprechenden Spalten des STEPPERS aufgerufen werden.

Mit der Demo, die zunächst nur extern verknüpft bleibt, können nun alle Wörter aus dem Grundwortschatz lautweise "zusammenbuchstabiert" werden. Die überschaubare Menge und Dateigröße dieses vorgenerierten Audio (knapp 40 MB) macht den später beabsichtigten Offlinebetrieb der App als PWA (Progressive, d.h. installierbare, Webapp) möglich. Ich hege die Hoffnung, dass der jetzige „working prototype“ tatsächlich als „proof-of-concept“ überzeugen kann, dass eine weitere Entwicklung und Förderung dieses Features lohnenswert ist (vergl. „Ideen für die Weiterentwicklung“)

PORTFOLIO

Sowohl Audioaufnahmen als auch geschriebene Buchstaben können derzeit als mp3 bzw. SVG/HTML ausgesichert und ggf. verschickt werden (beispielsweise um Andere mit besonders gelungenen Versuchen zu erfreuen oder die Erledigung von Aufgaben zu dokumentieren).

Die gespeicherten Dateien sind mit den Utilities und der Dateiverwaltung des Betriebssystems (Android) zu ordnen und zu öffnen.

Geschriebene Buchstaben („Spuren“) im Modul WRITE sind datenmäßig extrem kompakte, kleine SVG-Dateien (um die 1.5 kB pro char). Sie können, automatisch nach jedem vollständigen (Schreib)Versuch oder nach Entscheidung durch die NutzerInnen, als ein Frame („Einzelbild“) für eine Animation ausgesichert werden. Nach (beispielsweise) hundert so aufgezeichneten Wiederholungen, können die Frames, als „Zeitraffer“ abgespielt, den Lernfortschritt visualisieren.

Eine Demo (Homepage) .../alphasprint/EIPLAYER/click_to_play.html zeigt die "Gamifizierung" durch diese Funktion und illustriert m.E. wie hier Möglichkeiten bestehen, das UserInnen-Engagement zu binden.

Eine Schnittstelle für die rein statistische Dokumentation und Auswertung der Übungen (Wiederholungen pro Graphem) ist im Code angelegt, z.B. für zukünftige "spaced repetition" ("Wiedervorlage") Funktionen.

Konnten alle Meilensteine erreicht werden?

Eine vollständige Integration der drei letztgenannten Features konnte im Förderzeitraum nicht erreicht werden. Die Module sind aber extern verlinkt und können auch getestet und evaluiert werden.

USER INTERFACE

Das Ziel einer sich durch Ausprobieren weitgehend selbsterklärenden Bedienbarkeit/Nutzerführung („discoverable“, intuitive, self-explanatory) konnte m.E. im wesentlichen realisiert werden.

Die Auswahl des zu bearbeitenden Graphems (Buchstabe oder Buchstabengruppe) erfolgt durch Auswahl aus einer Liste, die Modalitäten der Bearbeitung (Lesen/Vorschrift, Schreiben/Nachspuren), finden sich im ersten Teil des oberen Hauptmenüs.

Hören/(Nach)sprechen ist als "Meta"-Funktion für den Wechsel zwischen Schrift- und Lautgestalt auf allen Seiten verfügbar (Fussbereich).

Die weiteren, "fortgeschritteneren" Features QUIZ (eine Variante von WATCH, für die

Wiedererkennung/Unterscheidung von Buchstaben/Graphemen anhand ihres Bewegungsmusters), STEPPER (Progression zu Lautfusion und Silben/Wortbildung), sowie PORTFOLIO (Dokumentation Lernstand) können aus dem zweiten Teil des Hauptmenüs aufgerufen werden.

Die „Ikonografie“ der Buttons sollte sich spätestens nach einmaligem Ausprobieren erschließen.

„ERKLÄRTEXTE“/MEHRSPRACHIGKEIT

Eine Präzisierung und Übersetzung der (unvermeidbaren) „Erklärtexpte“, die zur Zeit provisorisch nur auf Englisch vorhanden sind und nur pauschal bei Erstaufruf/Laden der App durch einen Pop-over-Dialog aktiviert werden können, in aktuell verbreitete/nachgefragte L1 (Erstsprachen) von NutzerInnen, erscheint sinnvoll.

Dies war allerdings in vollem Umfang im Rahmen des Projektes nicht möglich und nicht vorgesehen.

INTEGRATION

Die Module LAUNCHER, STEPPER UND PORTFOLIO konnten nur ansatzweise weiterentwickelt und eingebunden werden. Siehe dazu den Punkt (5) „TEXT-TO-SPEECH“, der aufgrund seiner Komplexität abgekürzt werden musste.

Auf die Einbindung in „Swiper“ (s. Punkt 5 FRAMEWORK) musste/konnte zunächst verzichtet werden.

Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Ich hoffe auf einen „viralen“ Effekt (in der Bekanntheit des Projekts) über die nachhaltige Listung des Projekts auf den Prototype Fund/OKF Webseiten. Hier bestehen zukünftig u.U. auch Möglichkeiten zur Kooperation/zum Austausch mit anderen BMBF/PTF geförderten Projekten im Bereich „Bildung“ (insbesondere „Lunes“).

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts?

Mit dem jetzt realisierten Prototyp kann Testing und Evaluierung unter „Realbedingungen“ von und mit UserInnen durchgeführt werden.

Ich hoffe auch, dass die App sich bereits in dieser Form (als „minimum viable product“) als hilfreich und nützlich für LernerInnen erweisen kann. (Github .../BLN_ABC)

Welche weitergehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse?

OPEN EDUCATIONAL RESOURCE

Alle oben näher beschriebenen Module/Features der (Proto-)App sind für das Selbststudium oder in oder begleitend zu Alphabetisierungskursen (Deutsch als Zweitsprache) nach einer minimalen Einführung einsetzbar.

Ich gehe davon aus, dass Module der App als ein Betrag zu OER auch für andere Anwendungen interessant werden können. Insbesondere wäre die (typo)grafische Übertragung/Anwendung auf

andere oder nicht-lateinische Schriftsysteme, von Walisisch bis Kana, relativ unaufwändig möglich. Dies könnte auch für enzyklopädische Anwendungen (Wikipedia) interessant sein. (Die phonetische Realisierung bliebe dabei allerdings zunächst schwierig, vergl. den nächsten Absatz „Ideen für die Weiterentwicklung.“)

Alle Features sind so modularisiert, dass sie auch unabhängig voneinander verwendbar/anpassbar und integrierbar sind. Da bewusst auf das Aufsetzen auf ein (Software-)Framework verzichtet wurde, liegt aller code in transparenter („vanilla“ JavaScript) Syntax vor und ist leicht zu isolieren. Zur Umsetzung sind, außer JavaScript/HTML/CSS, keine weiteren Kenntnisse erforderlich.

ANFANGSSCHRIFT

Mit der lizenzfreien „Anfangsschrift“ (Github [.../anfangsschrift](#)) steht ein „handschreibfreundlicher“ und ästhetisch und typografisch überzeugender lateinischer Zeichensatz, in einem einfachen skalierbaren Format (SVG/JSON) zur Verfügung, der leicht modifizierbar und erweiterbar ist.

TRACING-ALGORITHMUS

Die Implementierung für das interaktive „Tracing“/Nachspuren mit Korrektur-Feedback geht m.W. qualitativ deutlich über bisher verfügbare Ansätze (sowohl FOSS als auch proprietär) hinaus.

GRAFIK-PIPELINE

Die gut dokumentierte Pipeline für Sampling und Rendering und reaktives Nachspuren der Schriftpfade macht m.E. den Einsatz auch diesseits und jenseits von Deutsch/Zweitschrift vorstellbar, allgemein für Schriftrendering/Animation oder für schnelle Skizzen zum Schriftentwurf (als Export von geglätteten SVG-Pfaden).

REC(ORDER)PLAY(ER)

Code für das Modul RECPLAY wird zusätzlich mit generischem UI als vielseitig einsetzbare Utility bereitgestellt, mit der User-Audio aufgezeichnet und abgespielt werden kann (quasi wie ein „mini-whatsapp“).

Der code ist eine Erweiterung eines Tutorials auf MDN und implementiert die MediaStream Recording API. Weitere Anpassungen sind noch möglich für „scrubbing“ und Audio-Zeitleiste. (Home [.../alphasprint/RECPLAY_API/recplay.html](#))

LEXIKON

Das bereits vor dem geförderten Projekt (als Grundlage für den „Silbenschieber“) erarbeitete LEXIKON ist m.E. eine wertvolle Resource für Lehrende, um Übungsmaterialien nach verschiedenen „musterbasierten“ didaktischen Kriterien zu erstellen.

Das Lexikon umfasst einen (kuratierten) vertonten Wortschatz von 2.200 Einträgen, der mit Volltextsuche, durch Filter und beliebige, auch positionale Strings für An/In/Auslaut oder kontrastiv (z.B. „i“/„ie“, „ei“/„ie“) erschlossen werden kann (siehe Github [.../searchable_speaking_lexicon](#)) Die Datenbasis für das LEXIKON ist ein (schlichtes) Excel-Spreadsheet, das (mit Vorsicht) auch ohne weiteres editiert/ergänzt werden kann.

Der Wortschatz setzt sich aus dem Grundschulwortschatz NRW und Berlin/Brandenburg und handverlesenen Wörtern (bevorzugt die Verschriftung von „schematypischen“ Beispielen) aus DaZ zusammen, u.a. auch die „Anlautwörter“ aller einschlägigen Lehrwerke.

Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?

STEPPER („SILBENSCHIEBER“)

Das (oben beschriebene) LEXIKON ist die Grundlage für den STEPPER, der konzeptionell und didaktisch die „Brücke“ von Alphabet zu Silbe und Wort darstellt und die Progression von gesicherter Buchstabenkenntnis zu Semantik und (Grund)Wortschatz unterstützen soll.

Die Wörter im LEXIKON werden über einen "Stammbaum"-Algorithmus in eine treppenartige Hierarchie („Kladen“ <https://de.wikipedia.org/wiki/Klade>) zerlegt und gegliedert, mit der für jeden Buchstaben alle möglichen, d.h. im Wortschatz vorkommenden, nächsten benachbarten Folgebuchstaben („next neighbors“) angezeigt und aufgerufen werden können.

Für den derzeitigen Wortschatz (ca 2.200 Einträge) erzeugt dies, nach dem Muster „W“, „Wo“, „Wor“, „Wort“, ca. 8.500 Fragmente, die über TTS (Text-to-Speech) als Audiodateien synthetisiert wurden.

Daraus können alle Wörter schrittweise und akkumulierend, d.h. Buchstabe für Buchstabe, wieder (visuell und lautlich) „aufgereiht“ und angehört werden.

Dies kann als „Diktat“ eingesetzt werden — ein Wort wird vorgesprochen und muss aus den verfügbaren Buchstaben/Graphemen (re)konstruiert werden — oder umgekehrt als „Vorleser“, wobei ein geschriebenes Wort buchstabenweise, akkumulierend gesprochen wird.

Wie im realen (begleiteten) Unterricht können Lernende also das Lesen und die Aussprache von Wörtern durch lautliches „Zusammenziehen“ erlernen.

Dabei ist die Absicht keineswegs, den gesamten Wortschatz zu vermitteln, sondern zu ermöglichen, das Vokabular frei (oder durch Aufgaben gelenkt), „spielerisch-explorativ“ durch-forschen und untersuchen zu können und dabei augenfällige und hörbare Schemata und Muster zu entdecken.

Ziel ist also „ein Gefühl für die Regeln“ (d.h. Intuition) zu entwickeln, nach denen die Wortbildung und Verschriftung im Deutschen erfolgt.

Gleichzeitig können auch „pragmatisch“ interessierende Wörter entdeckt und gelernt werden, die helfen „semantische Inseln“ in der fremden Sprache zu etablieren.

Der STEPPER wäre ein m.W. bisher nirgendwo realisiertes Feature für eine Alphabetisierungsapp, das den Einsatzbereich und damit den Lernerfolg erheblich vergrößern könnte.

Meine Erwartung wäre, dass sich mit diesem Tool der Erwerb der „magisch“ anmutenden Fähigkeit zum kognitiven Verarbeiten von (Schrift)Zeichen als (Sprach)Klang produktiv „anschieben“ lässt.

Um automatisch einen Wortschatz wie im LEXIKON, der editierbar und erweiterbar sein soll, auch mit verschiedenen Stimmen (vorrangig männlich/weiblich) synthetisieren zu können, ist wahrscheinlich das Training und die erhebliche Modifikation eines "custom" Phonem-Modell erforderlich (möglicherweise basierend auf „diphon“ und/oder „formant synthesis“, vergl. <https://github.com/espeak-ng/espeak-ng>).

Es bleibt vielleicht zu klären, inwieweit diese Pipeline dann u.U. auch auf andere Sprachen übertragbar wäre oder als Infrastruktur/platform/stack für ähnliche Anwendungen interessant sein könnte.

Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Die ergebnisorientierte Arbeit am Projekt erforderte ein mehrmaliges „Nachstellen“ des Zeitplans und der To-Do-Liste.

Ohne die Reflexion und Strukturierung durch die regelmässigen „Montagsupdates“ und die späteren Coaching-Einheiten bei Superbloom wäre einiges aus dem Ruder gelaufen und hätten wichtige Teile komplett untergehen können (z.B. Icon-Entwicklung).

Die Arbeit am Projekt hat mich einer pragmatischen/realistischen Einschätzung des Aufwands und Zeitbedarfs näher gebracht (vergl. https://en.wikipedia.org/wiki/Ninety-ninety_rule).

Natürlich konnte/musste ich meine JavaScript-Kenntnisse vertiefen und anfängliche Annahmen zu Design und Usability überdenken oder konkretisieren.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?

TEXT-TO-SPEECH

Die automatische Vertonung der einzelnen Fragmente mit TTS hat sich bei den derzeit dominierenden "state-of-the-art" (deep learning) Synthesemodellen als Haupthindernis für die Umsetzung herausgestellt.

Der Hauptgrund dafür dürfte sein, dass in den Trainingsdaten dieser Modelle die oben beschriebenen "Fragmente" nicht, jedenfalls nicht isoliert, vorkommen. Mit anderen Worten: das Syntheseprogramm kann auch nicht "sprechen", was es vorher nie "gehört" hat.

Auch Transkription der zu sprechenden Texte in SAMPA oder SSML war für isolierte Phoneme nicht wirksam. In allen Fällen fingen DNN TTS-Systeme an, unterhalb von 3-4 Buchstaben (auch bei zusammengesetzten Grapheme wie „sch“ oder sprechbaren Silben wie „ba“, „be“ ...) zu buchstabieren, d.h. die BuchstabenNAMEN aufzuzählen, statt zu den Lautwert zu sprechen

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?

„SWIPER“-FRAMEWORK

Die umfassende Einbettung aller Module (und des gesamten Übungsinventars) in ein „swiper“-basiertes Navigations-Framework wurde zurückgestellt.

(„swiper“ ist die führende Open Source Lösung für GUI/UIUX, die für die Navigation in touch-fähigen Apps hauptsächlich auf die User-Geste „wischen“ aufbauen. Siehe Github [.../nolimits4web/swiper](https://github.com/nolimits4web/swiper))

Der Einsatz von „swiper“ hätte eine weitere Verkapslung der mittlerweile recht gestiegenen Komplexität des Programmcodes erfordert.

Zudem war meine Implementierung für die Unterscheidung von „wischen“ mit 2 Fingern – zum

Navigieren/Blättern – und „wischen“ mit einem Finger – für das Schreiben/Nachspuren – nicht wirklich zuverlässig. Mithin hätte das ein Risiko für die (auch Plattform/Browser übergreifende) Lauffähigkeit der App bedeutet.

Die Navigation konnte, problemlos und ohne Verlust von Funktionalität, auf Basis eines „konventionellen“ UI (Blättern mit „vorwärts/rückwärts“ Schaltflächen und Auswahl der Programmfunktionen mit „normal“ klickbaren, kontextuell aktivierten Icons/Buttons) umgesetzt werden.

Nichtsdestoweniger würden die Gesten in „Swiper“ für einen weiteren Ausbau der App die „eleganteste“ Umsetzung und eine tatsächlich noch intuitivere („touch-native“) Navigation zwischen Buchstaben/Buchstabengruppen (swipe links/rechts), Klein/Großschreibung (swipe rauf/runter) und Bearbeitungsmodus darstellen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GHitHub, Veröffentlichungen)?

Die Ergebnisse (d.h. der vollständige source code) sind über mein Github Repository github.com/eieye/BLN_ABC verfügbar.

Eine Demo/Installation der App ist verlinkt und kann von dort gestartet werden. (Die App wird in einem „secure context“/https gehosted, damit die User-Permission für das Audio-Recording erfragt werden kann.)

Weitere Demos sind über QR-codes von meiner Homepage unter <https://www.jenskreitmeyer.de/alpha/beta/demos.html> abrufbar.

Der code für die gesamte App ist (durch JavaScript exports/imports) sehr stark modularisiert und auch bewusst ausführlich(st) kommentiert belassen, so dass einzelne Funktionalitäten (Features) relativ leicht zu isolieren und nachzuvollziehen sein dürften.

Das README im Repository erläutert die hauptsächlichen Absichten hinter dem Projekt. Weitere Informationen über die Entwicklung/Vorgeschichte sind auch über meine dort verlinkte Homepage auffindbar, u.a. der didaktikbezogene Theorieteil meiner BA-Arbeit (https://www.jenskreitmeyer.de/app_ba/BA_Theorie_20190728.pdf)

(Domain-Pfad in diesem Bericht für Github: [https://github.com/eieye/...](https://github.com/eieye/) für Homepage: [https://www.jenskreitmeyer.de/...](https://www.jenskreitmeyer.de/) in Arbeit: www.blncbc.de)

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

AUDIO-VERVOLLSTÄNDIGUNG

Die erforderlich gewordene „mündliche“ und manuelle Überarbeitung bzw. Ergänzung der Audio-„Schnipsel“ aus der Text-to-Speech Sprachsynthese hat unerwartet viel Aufwand (ca. 4 Wochen)

erfordert, die z.T. auch auf Kosten anderer Teile gegangen ist: nicht wirklich quantifizierbar, aber anzunehmen bezüglich der Hilfe/Erklärtexthe, der Anbindung des LAUNCHER (Vertonung der Tastatur) und mehr Optionen für Einstellungen/Konfigurierbarkeit.

NOCH KEINE PWA

Zur Zeit ist die App noch nicht als PWA installierbar. Die App kann aber ohne Einschränkungen aufgerufen und genutzt werden, allerdings im Moment nur bei einer aktiven Internetverbindung. Aufgrund des kleinen „footprint“ von 3 MB für den „app core“ mit dem vollständigen deutschen Phonem/Grapheminventar, plus (optional bei Aufruf) 36,3 MB für den „Stepper“ mit dem Audio aller Wörter und Fragmente, ist der (ggf. mehrmalige) Download aber eher unbedenklich.

Die Entscheidung für die Weiterentwicklung des STEPPER, zumindest bis zur, in der jetzigen Form, vollständigen und funktionierenden Demo als Proof-of-Concept, hatte für mich wegen des Innovationspotentials dieses Features klare Priorität (vergl. Punkt (4) „Ideen für die Weiterentwicklung“).

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Während der Projektlaufzeit sind mir keine derartigen Ergebnisse oder Entwicklungen bekannt geworden. (Andere Einflüsse sind ausführlich in meiner BA-Theoriearbeit dargestellt, s. Link oben)

Erwähnenswert sind allerdings die Herausforderungen und Hindernisse, die sich (eher unerwartet) aus der Schwierigkeit ergeben, mit allgemein zugänglichen DNN-trainierten Sprachmodellen phonetische, „sub-lexikalische“ Einheiten zu synthetisieren (Punkt (5) „Text-to-Speech“).

*Dank an BMBF, DLR-PT, OKF, Team Prototype Fund und M.W. Superbloom und K.K. und L.B.
Berlin, 08.04.2025*