LEUPHANA UNIVERSITY LÜNEBURG

TBII REPORT
SUMMER SEMESTER 2024

Module: Technological Basics II
Seminar: Tech Basics II - Stream A
Lecturer: Sarah Haq

WORT
SCHATZL

An application to celebrate the beauty of words by expanding the active vocabulary of users following a gamification approach.

submitted by

Max Schwendemann

Date: 28th June 2024

Table of contents

1 Introduction: Why *Wortschatzl*?

The idea behind *Wortschatzl* is simple: I want people to learn, remember and celebrate stunning, wonderful words in a joy sparking, playful and interactive way. Thus, with the creation of the *Wortschatzl*-idea I decided to send the users on a little exploration journey to discover German words that are simply beautiful.

The basic concept of the app is simple but appealing: the user gets two words randomly picked from the ‘treasure chest’ called *Wortschatzl* (a word list in a text file) and must create a sentence that contains these two words. Thereby, the user learns new words or remembers words he*she might not use a lot. This helps to broaden the active vocabulary and should also offer a pleasant moment in our often stressful everyday lives. Moreover, as a more general goal, it helps keeping words from being forgotten.

But that is not all there is: With *Wortschatzl* users can also have a look at and scroll through the complete word list, e.g. if they are searching for nice acronyms, letter starters or try to find nice words for similar approaches. And it should not only be my words that can be found in the treasure chest: I want to give the users the ability to also include own words via the app.

2 Methodology

So far, the application consists of two main elements: The python file called ‘*app.py*’ and the text file called ‘*worte.txt’*. To make the GUI more accessible and aesthetically pleasing, in the folder ‘*image*’ there are several images used as backgrounds for the GUI. Furthermore, the folder ‘*src*’ includes a phyton code called ‘*helpers*’ that stores functions that are needed more often and thus helps keeping the *app.py* file clean.

The *app.py* file can easily be run in the terminal. When executed properly, it opens the GUI showing the homepage which contains three main buttons that each lead to a subpage of the app. Depending on which button you choose, you can:

- Start you learning journey immediately by getting two nice German words you have to build a sentence with. Every time you get the spelling right, you get a coin. The score is displayed on the upper right hand corner of every page. All in all, the practice should help that the words stick in your head.

- Have a look at the words that are already part of your word treasure chest.
- Add new words that inspire or amuse you to the *Wortschatzl* so you can learn and remember them later.

3 Design

In TBI, the first thing to start with regarding the design of the app was the naming as I wanted to build the design narrative around it. I quickly decided to go with the term *Wortschatz* or at least to use it as a basis, because it is a beautiful word itself and offers lots of possibilities in regard of the design, e.g. a pirate treasure-hunt universe. Moreover, it also inhabits the idea of a rich vocabulary one can aim for. I decided to go with *Wortschatzl* giving the name a playful and cute character and sound. It also hints towards the possibility to include dialect words. As a next step, I designed a playful matching logo for the app and wireframes using the tool *Canva* (see title page).

With the basic design concept being finished, this semester I started to build the GUI itself. Here, I focused on a very simple layout and tried not to overload the screen with icons and information. The main goal was to create a very simple GUI to get an intuitive user experience. To make it optically appealing, I used a modern, but rather calm colorful background and some graphics related to the pirate theme. In the upper left, the lettering logo is always displayed. In the upper right, the coins are counted next to the treasure chest figure. The user always can go back by clicking on the map icon on the lower right (except on the homepage). To close the GUI, the user can press the skull icon on the lower left.

See a video of the working GUI here to get a better idea of the design: https://drive.google.com/file/d/1Htwx54VWbvZ56RuPxtqAJ4kScRp9m3hH/view?usp=share_link

To achieve a scrollable list on the wordlist page, ChatGPT was used: see line 25-32.

4 Limitations & outlook

As my mother tongue is German and this is the language I am by far most fluent in, I have the closest emotional connection to German words including some Badian dialect terms from the Middle Black Forest. This is why the text file currently contains German words only. To make the GUI accessible to the lecturer English was used for the GUI design. Nevertheless, the concept could easily be transferred into other languages if a database in form of a wordlist from the respective language is created and the GUI would be adjusted respectively.

Currently, there is one flaw in the code I was not able to fix yet which is the sorting of newly added words. Unfortunately, bringing the words in an alphabetical order does not include the words already given in the text file right from the start.

Moreover, the created sentences on the learning-journey-page can only consist of the two words and still gets accepted. Here, it would be great to proof whether the sentence shows certain quality standards. Furthermore, one could also think of ways to capture the created sentences and to reuse them at some later point in the game. Another idea is that users can ask for new words if the proposed two are not appealing to them or if their meaning is unclear. Of course, having the possibility of getting a proper word explanation with suitable examples within the app would be an even more professional feature.

And there are far more ideas to be realized that are missing still: In the long run, users should be able to log in and create a personal account to make user tracking besides the coin score possible. Furthermore, users should not only be able to add words but also to vote for or against submitted words of others. Only if a submitted word gets more than x upvotes during time period y the word gets included into the database. Besides that, users should also be enabled to use their coins to shop in a pirate shop. There, they might use them for different advantages, e.g. doubling the amount of coins collected in a set time frame. It would also be nice, if one could unlock new areas of a map where a pirate ship could sail to and thus explore new word-worlds. One could also think of gaining different piracy levels or creating an own pirate character you can buy new clothes, accessories or ships and flags for.