communicated by:
Lehr- und Forschungsgebiet Informatik 9

Prof Dr. Ulrik Schroeder

RWTH AACHEN UNIVERSITY

The present work was submitted to the Learning Technologies Research Group
Diese Arbeit wurde vorgelegt am Lehr- und Forschungsgebiet Lerntechnologien

Crossword Puzzles in Digital Education with Gamebooks
Kreuzworträtsel in der digitalen Lehre mit Gamebooks

Bachelor-Thesis
Bachelorarbeit

presented by / von

Müller, Silvia Alexandra
445514

Examined by / Begutachtet von

Prof. Dr.-Ing. Ulrik Schroeder
Dr. Svenja Noichl

Aachen, September 8, 2025

Contents

List of Figures

Abstract

One version of digital education is gamebooks that offer the user several possible learning paths through the given learning materials using gamification. This work proposes an implementation for crossword puzzles as a new task type for the platform DiGaBo which enables teachers without programming knowledge to create their own gamebooks and offers a platform for learners to play them [SN21]. Many research papers show the positive effects of serious games that use gamification for educational purposes [LP19, IZ22].
The research questions of this paper are what makes crossword puzzles valuable for education and how this task type can be implemented. This also includes comparing generation algorithms, discussing the needed functions for users and the visualization of the puzzle grid.
With the proposed solution, a teacher can generate a crossword puzzle from input clue and answer pairs and include it into a created gamebook. The algorithm for the crossword generation is strongly influenced by the algorithm of Agarwal and Joshi [CA20] but for this use case, weaker conditions than in their case apply. Additionally, randomization is used. The learner can fill in these crossword puzzle within the gamebook and receive visual feedback and hints. Since crossword puzzles are one of the task types that increase the motivation the most[LG19] and their positive impact on the learning process can be confirmed empirically [AA24, NK22, EKC83, Msh20], they are a valuable extension for the platform and can improve the learning process if well-thought-out clue and answer pairs are used.

Chapter 1 Introduction

Overview

Crossword puzzles are a popular and challenging puzzle type that especially appears in newspapers. This work has the goal to include this puzzle type as a task type into the platform DiGaBo which enables creating digital gamebooks for teaching purposes. This paper explains why crossword puzzles improve the platform for learners as well as teachers and give an overview of the positive impacts of this task type on the learning process. The chosen implementation using Angular, JavaScript and node.js as well as the chosen crossword generation algorithm will be explained. Furthermore, this paper will point out important factors for the use of the new task type.

1.1 Motivation

The Platform DiGaBo is a learning technology that uses gamification and active learning to impart knowledge. It already offers several task types for greater variety and enjoyment during the learning process.

Active Learning

The goal to use active learning and gamification bases on the opportunities they offer. Own actions can increase the percentage of the learned information to 75 % according to the learning pyramid by the NTL Institute [Let12]. In general, active recall after a passive learning phase is more efficient because of the testing effect. This effect is reinforced if there is a test after a longer break and if feedback is provided [lex, RPS11]. With this knowledge, gamebooks can be a great tool to create efficient learning material. Furthermore, gamebooks offer the opportunity of individual learning paths and can include more help and repetition tasks for students in areas where they have less knowledge. Additionally, the media competence might be increased since this multimedia platform is used on an electronic device. There are also the advantages of a digital crossword puzzle that offer easier crossword generation, the use of multiple layouts for the grid and instant and individual feedback.

Gamebooks

Media Competence

variety

The goal of this work is to offer a new task type not only for wider variety but also to increase the understanding of the learning material because of the puzzle character. Crossword puzzles can be used in many different ways, like for vocabulary training, but also with more difficult clues that require a deeper understanding and more complex thinking. Referring to the CELG-Taxonomy (Computer Supported Evaluation of Learning Goals) table that distinguishes between different levels of knowledge and different levels of cognitive processes, it is possible to assign tasks to fact knowledge, concept knowledge and procedural knowledge and also to reproduction, understanding/application, reflection/evaluation and creation [Lem14]. The DiGabo Platform already offers many possibilities to cover these areas with its different task types.

So for example, a single-choice task can be used for reproduction, a Drag-and-Drop task can be used for the understanding/application dimension and a multiple choice quiz for reflection and evaluation. But the task types can often not only be used in one category, but depending on the way the task is designed, they require different cognitive levels [SN23]. Due to the different use cases of crossword puzzles, they can also cover multiple areas. They can be used to check all types of knowledge as reproduction ('English word for: Stuhl(German)', understanding/application ('What is essential for photosynthese? - answer: chlorophyll') or also evaluation('least efficient sorting algorithm') depending on the types of clues.

use cases

The most important part of gamebooks for active learning is the intended gamification. Currently the DiGaBo Platform is used for voluntary courses for seniors where the focus is more on the learning process and fun than reaching the end of the gamebook or achieving a grade. Here intended gamification is crucial and leads to an increased knowledge retention but this effect depends strongly on the quality of the learning material [Zho19, LP19].

Gamification

More specific reasons for crossword puzzles as a new task type that follow with the strong focus on gamification are the puzzling process, the included feelings of delight and success that comes with challenging clues and the help option that avoids frustration.

puzzling

A study shows that crossword puzzles are, with marking and matching tasks, one of the task types that inspire the most motivation [LG19]. Also their general popularity never decreased since their introduction in 1913 and they are used in many variations [McK, CA20].

motivation

Furthermore, it can be hard for teachers to create a puzzle without using irrelevant words just because they fit well into the grid. Whereas, using the most important key words for learning and repeating knowledge [LG19] would be more goal-orientated. To avoid this, the goal of this work is to create crossword puzzles from clue and answer pairs the teacher gives as an input so it is easier for them because they can focus on content instead of visual aspects. Additionally, it is simpler for the educators to have this function in the same platform as the other task types so it is included easily in the resulting gamebook. To ensure that the feature is easy to handle, the design will focus on intuitivity and simplicity.

easier generation

1.2 The New Feature

The goal of this work is to provide a new functionality in the current DiGaBo Project to use crossword puzzles as a learning format. Therefore, a possibility to create the crossword for educators without technical knowledge or programming skills is added in the editor. They are able to give clue and answer pairs as an input to the editor which generates a matching crossword that uses all answers. Furthermore, a preview is visible to regenerate the grids or to delete specific ones. Moreover, a solution word can be chosen optionally.

input in editor

The crossword itself is generated in the backend in a service using an algorithm that is inspired by the algorithm of Agarwal and Joshi [CA20] and that uses randomization to provide different grids that can be chosen in the editor.

generation in backend

Additionally, the crossword task type is integrated into the user interface (UI) for learners so they can fill in the crossword puzzle with the possibility to use a help option and to perceive visual feedback. The evaluation on the user's solution is also saved for further analysis of the gamebook.

puzzle in UI

The main goal of this feature is to offer the functionality to create, edit and play digital crosswords in the DiGaBo platform. This is done in a intuitive and simple way that helps the users to learn the included knowledge and intensifies the enjoyment during the learning process. Since the DiGaBo platform is mainly used on tablets or laptops the view is not primarily designed for smaller devices like mobile phones. Furthermore the proposed solution assumes that the functions will be used as intended.

main goals

Chapter 2 Related Work

To design and implement this new feature, it is important to understand previous research results and related projects. Knowing why active learning and the use of crosswords is beneficial helps to clarify what the important factors for the successful use of crosswords are and how they could be designed. Furthermore, general knowledge about the platform DiGaBo is essential since it defines the use case. This work is an extension for it and therefore has to be integrated smoothly. But the most important parts of the related work are existing crossword generation algorithms and their differences as well as visualization examples. *overview*

2.1 Active Learning

Since the learning scenario is part of active learning, it is important to understand the positive effects of this learning strategy and to identify crucial factors.

There are several definitions that show the covered range. So active learning refers to all scenarios where the learner is actively involved in the process and is able to influence the learning process by choosing the order and the speed of the learning material [BO22]. There is evidence that active learning can improve the memory performance of adults and children [BO22]. Especially the feeling of self-determination is important [AR19]. *definition*

From the perspective of cognitive psychology the better encoding into the episodic memory, that deals with personal experiences and is part of the long-term memory, can be explained on one hand by the additional stimuli during execution. On the other hand, the purposeful decision-making in the planning phase requires a mental representation of the knowledge and therefore leads to better encoding. *cognitive factors*

Additionally, the learners can choose the speed of the learning process and therefore more effectively use their own selective attention which leads to better memory performance. Another reason for that is the adaptive selection of learning material because the learner can choose the material that fits best to the individual knowledge level.

Also the term of metacognitice monitoring is mentioned, that refers to monitoring own progress as well as the own knowledge level and increases the effectiveness of the learning process [DB16, BO22].

2.2 Advantages of Crossword Puzzles

Even if there is not much research for crossword puzzles as part of digital gamebooks, they are part of active learning and there is a lot of research of crossword puzzles in

education.

positive effects for all ages

There are many studies in different areas and with participants of all ages. Often they are used to learn and remember terminology but also for a higher level of thinking [Msh20]. Both in aiding English language learners as well as in learning medical terminology and in pharmacology crossword puzzle improved the vocabulary knowledge even if the terminology is considered difficult to recall [SP18, Mer16, AA24].

learning terminology

Also in nursing education, undergraduate courses in food science and technology as well as in courses about the history of psychology courses positive affects could be confirmed [NK22, Msh20, EKC83]. Even in a meta analysis including various studies in elementary schools in India when the puzzles were used for Social Science learning the benefits and the enjoyable learning conditions including the positive feeling of success and increased engagement could be shown [Fid].

increase engagement

Often these educational crossword puzzles were analog and themed [Mer16]. Besides the increased motivation and a generally positive experience, the students also felt requested to think critically and saw the puzzles as an opportunity to asses their level of understanding of the topic [Msh20]. Furthermore, many where also already familiar with this puzzle type and therefore more interested [EKC83]. When they were used as self-learning tool in pharmacology, a confidence boost for the learner also emerged [SP18].

critical thinking

Since even more complex knowledge than vocabulary can be practiced, there are also approaches to use them in education of computer engineers within an educational game increasing the knowledge of basic terms and concepts in C/C++ programming [ABA24].

Especially for introducing new vocabulary, testing vocabulary, teaching spelling, improving analytical and cognitive skills as for relieving stress during the learning process and for a feeling of accomplishment within the learners, crossword puzzles are a useful tool and offer more opportunities than the closely related wordsearch puzzles. Pedagogical suggestions also recommended to integrate them as warm-up or wrap-up activities or as an enjoyable vocabulary test into lessons [AA24].

suggestions for the usage

72 % of the Students that participated in a multidisciplinary course using e-crossword puzzles also stated that the puzzles were more interesting for them than other activities such as quizzes or tests [STN23].

In addition, cooperation between learners could increase the success and avoid frustration [Mer16]. Because even if the positive effects appeared, there were also negative feeling towards the puzzle like the anxiety not to be able to solve them because of a lack of knowledge, a lower skill level and time constraints. There were also students with a general dislike for puzzles and feeling of wasting time [NK22]. Especially for these student, a gamebook could offer an alternative path so that they can learn without the puzzle type they dislike.

negative feelings

Nevertheless, an online study could show that crossword puzzles as well as marking and assignment task motivate learners the most, compared to Single/Multiple Choice tasks, ordering tasks, fill-in-the-blank and free text tasks. The interest is higher and considering crossword puzzles, the participants perceived their chance for success as better.

motivation

In general, the motivation depends strongly on personality and the current situation but it can be shown that learners have preferences towards different task types. But nevertheless, the use of multiple task types is desired and especially crossword puzzle are presented as a good choice for digital education [LG19].

gamification

The most important factor for the increased motivation that is the gamification of the learning topics. The crossword puzzle type offers a challenge for the learners with the goal of filling the grid and finding the solution word. Since the effectiveness of gamification for the learning progress were confirmed in many papers [IZ22, LP19], this is a strong advantage of this task type.

crossword specific hints

Another advantage of crossword puzzle is, that they offer more hints than other quiz types to solve the puzzle than just the clues. Since it is a knowledge-based activity often people are unsure if their answer is correct and might not fill the word into the gap, but when there is an intersection with another word that reinforces the assumption, there confidence is strengthened and the gap is more likely to be filled. Furthermore, it might be possible to recognize answer words even if the person would not be able to produce these on their own so the knowledge can be strengthened. Moreover, while more specific clues might be more effective, also less frequently used letters in the grid are more likely to obtain more information because they do not occur in many words and therefore are more useful for the user [Nic11].

choosing clues

One criterion for good vocabulary exercises is the usefulness of words, so interference should be avoided by not including related unknown words [Mer16]. When the crossword puzzle is used for vocabulary training, this applies as well.
To create solvable crosswords, it might also help to reduce cultural references so that the solution does not depend on the knowledge of a specific group of people [Mer16].

aesthetics

Additionally there are a few criteria for aesthetics that apply to crossword puzzle. Besides the visual appeal of the grid including symmetry and a certain visual form, the experience of struggling and finding the right solution is of interest. This is also part of other puzzles types and can be perceived as aesthetic. Here the challenge at the limits of one's own intellectual abilities and knowledge is important to make positive emotions of the challenge prevail. Therefore cryptic clues should feel clever and economical and the overuse of unnatural words should be avoided. Furthermore there are non-aesthetic aspects like morality and politics that typically influence the crossword creation. So unsightly words like 'rectal' as well as offensive clue-answer pairs are avoided by the New York Times. Moreover, there is a focus on diversity of pools for clues and an expand range of cultural references [Kub23].

2.3 The Platform DiGaBo

Digital Gamebooks

The Platform DiGaBo is a platform for interactive multimedia digital gamebooks which supports active learning. These gamebooks give the learner more freedom in decision-making and enable nonlinear and individual learning paths. The interaction helps to design a more diverse, captivating and efficient learning process by covering

all areas of the CELG-taxonomy [SN23] if it is used in a well-designed gamebook. It can also provide individual support and allow computer-supported evaluation of learning goals.

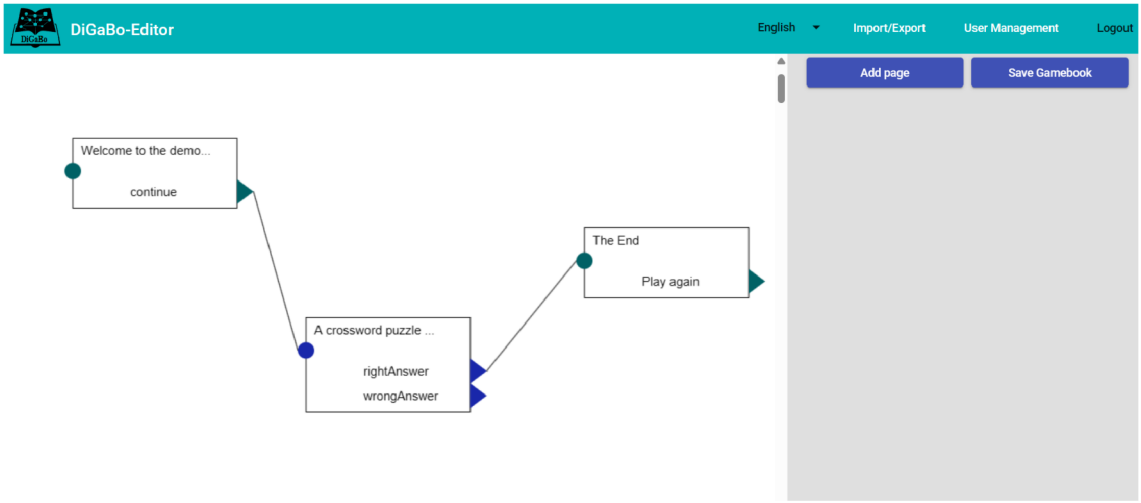

Figure 2.1: Screenshot of the DiGaBo Editor

Editor

One website of this platform is the DiGaBo Editor and it has the goal to enable all teachers after a short login to create their own gamebooks without any programming knowledge. The Gamebook is represented as a graph that uses nodes for every page with content and edges that represent connections between them and show which paths are possible to play. An example can be seen in figure 2.1. Each node offers the possibility to write down information or task instructions as well as multimedia content. The teacher also has the option to choose between the given task types. These are currently Choice, Sorting task with two or three categories, text input, Multiple Choice, Order, Matching Pairs, Rating Scale, Number Scale and due to this work Crossword [SN21]. The gamebooks can also be imported and exported [1].

task types

UI

The created gamebooks can be played in the user interface, which is the Website called DiGaBo. The user has to log in and can then choose a public or self-created gamebook to play. It is also possible to change the language or the visual theme and log data can be displayed and saved [2].

2.4 Crossword Algorithms

Knowing the framework conditions, there are multiple possibilities to implement a crossword generation algorithm.

In California, Matthew Ginsberg investigated several factors that are crucial for crossword generation algorithms like a form of look ahead and the amount of information during running time. The importance of good heuristics is obvious since already with a 4 x 4 grid using a large dictionary brute force search is impractical [MLG90].

difficulty

[1] https://editor.digabo.elearn.rwth-aachen.de, version from 08.04.2025
[2] https://digabo.elearn.rwth-aachen.de, version from 08.04.2025

The following examples represent some existing approaches for crossword generation algorithms.

CLP

One heuristic to generate grids uses logic programming. In 1997 an implementation using C and Prolog was compared with another that used the constraint logic programming (CLP) system CHIP with a C language interface. They used the word-by-word and the letter-by-letter-approach. The efficiency of the word-by-word approach relied on 3 aspects: the choice of which pattern (variable) to fill, the choice of a suitable word (value) form a large database and the backtracking strategy. To maximize the number of choices for the remaining words in the grid, the most constraint pattern is instantiated. Prolog also uses simple backtracking while CHIP has a more efficient backtracking strategy [GM97]. So logic programming can be used successfully to solve the problem of creating crossword puzzles.

Constraint Satisfaction Problem

In Italy another approach was applied resulting in a complete software system with the ability to create crosswords including the clue generation using natural language processing. The grid generation is done by a Constraint Satisfaction Programming solver. To use this solver the crossword rules are represented by constraints. The grid gives layout constraints that keep the placed words in the boundaries of the grid while schema constraints are defined for previous inserted words and therefore are introduced during the creation process. The solving system works with scores for partial solutions to measure how close they are to a full solution. The researchers showed that there software was able to often create a full solution or if that was not possible a very close one within a maximum of 10 000 iterations using a set of 91 000 definitions that were also generated by the system [LR08].

another implementation

There is also an implementation in python on GitHub by Otis Peterson and Michael Wehar that uses constraint satisfaction by calculating all possible words for every gap and then filling in the gap with the least options before recalculating the possible words again. This is done until one gap has no more possible word because then backtracking is done that uses look ahead [MIm].

using relative coordinates and concatenation

But besides logic programming and constraint satisfaction there are also other approaches. In Poland, dynamic slot tables were used for the storing of the data so that smaller crosswords were saved as relative coordinates which made them easier to concatenate or rotate. They used a best improvement search as well as a combination of first and best improvement local search. Their research also included a goal function that is calculated as $intersections \cdot \frac{letters}{width \cdot height}$ [JD22].

algorithm of Agarwel and Joshi

Another crossword generation algorithm was proposed by Charu Agarwal and Rushikesk Joshi in India. The algorithm uses given clue and answer pairs and assumes that there is no fixed grid size or geometry given (also called unconstrained crossword puzzle generation problem). The increase in computational complexity with the count of words is linear and also clue generation can be added to complete the end-to-end algorithm. The grid generation algorithm itself combines multiple strategies. The initialization includes choosing a subset of the given words and setting the grid size to then length of the longest. Additionally, the chosen words are

ranked ascending based on the number of intersections with other words.
Then words are placed, starting with the first word in the center. The order due to the ranking is called Pebbles and Sand principle because the difficult words are inserted first just like filling a jar with pebbles and sand. When no word is placeable, one word that enables the insertion of more words possible is removed (Greedy Removal Strategy) and if there is no such word, the most recently word is removed (Last In First Out). The greedy removal strategy can also result in disconnections that need post-processing to find connecting words. For danger of cyclic removals, that occurs especially in smaller data sets, Agarwel and Joshi suggest reducing priorities of words once backtracked.

pebbles and sand principle

During this process the best grid so far is stored and the process is repeated with a larger grid. As a termination condition, an upper bound on the number of iterations is used. Agarwal and Joshi also investigated how different algorithm design decision influenced the performance to find the best strategies and prove the effectiveness and scalability of their algorithm [CA20].

Since the focus of this thesis is the use of crossword puzzles for educational purposes focusing on the integration of this new task type into the presented DiGaBo platform, many of the approaches above are too complex and less effort for generating a crossword is needed with a smaller set of words and also weaker constraints for the resulting grid.

2.5 Visualization

If the crossword is created, there are different options to visualize it. The main difference is the representation of the blocked cells. They can be filled in black (as shown in figure 2.3) so that the puzzle is perceived more as a squared grid [CA20] or they can be omitted (as in figure 2.2) so that the crossword puzzle appears more fragile.
When creating a crossword with the focus on vocabulary learning often this gridless option is chosen [Mer16] but the black square format is often not associated with studying or school so that they might be perceived as more enjoyable [EKC83].

grid

clues

The description with the clues for the answer words is often displayed as a table with one column for horizontal and one for vertical word (also as across and down). It is also common to use small numbers for the white cells that represent word beginnings [CA20].
Furthermore, a solution word appears in several crossword puzzles and is often represented by small letters and highlighted cells using a background color or a circle [visa, visb].

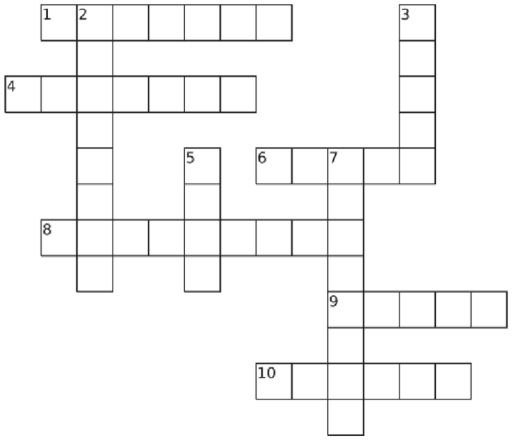

Figure 2.2: Example for a gridless crossword puzzle (excerpt from [Mer16])

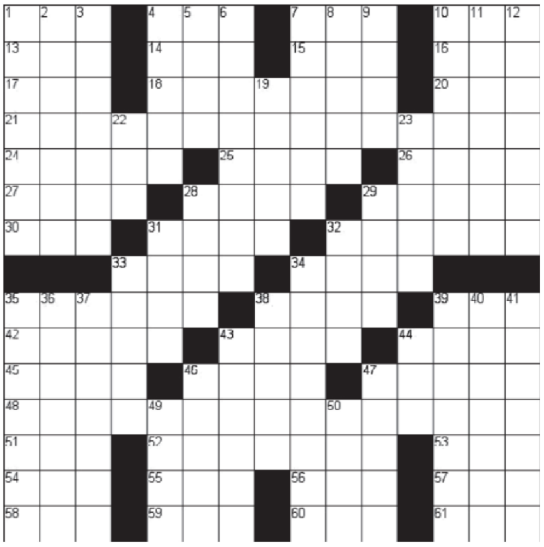

Figure 2.3: Example for the visualization using black squares (excerpt from [Mer16])

Chapter 3 Foundations

overview

Besides existing research and approaches there are important foundation to consider. The way how crosswords are created and what characteristics they can have influences the decision-making in the implementation process. Furthermore, there are design guidelines important especially for the visual aspects and technical foundations like the technologies of the platform DiGaBo and the RegExp object in Java Script for regular expressions as well as the structure of Angular that are relevant to understand the implementation.

3.1 General Foundations

3.1.1 Crossword Creation

crossword concept

A crossword puzzle is a grid with white (empty) and black (blocked) fields. The white cells must be filled by the user to solve it. Here every sequence of white fields has a number that matches to the list next to the grid where the user can find the clues that describes the fitting word and gives its orientation (horizontal or vertical also often referred as down and across). Often a crossword also includes a meaningful solution word that is a compound of certain letters in the grid. This implementation also offers the option to choose a solution word in the editor.
The size of the crossword grid is typically a square and sometimes the black fields build a certain form. In literature, the term vocabulary is often for a database of question answer pairs dictionary, but commonly, it is not necessary to use all of them in contrast to answer words in this thesis.

cryptic vs. non-cryptic

Besides themeless and themed crosswords, there are two different types of crosswords. The cryptic ones (also called British) and the non-cryptic (American) puzzle. The main difference is the type of clues that are given for the answer words. While non-cryptic crossword puzzles contain only straight definitions and require knowledge of vocabulary and cultural context, cryptic puzzles have besides the precise straight definition a fair elliptical indication of the answer which often increases their complexity in respect of solving. Additionally, they often give an enumeration of the answer's length in brackets after the clue [Kub23], which this implementation does not do. In general, both types of crossword puzzles are possible in the proposed solution of this thesis because the clues can be chosen arbitrary by the teacher, but for learning purposes non-cryptic puzzles are the main use case.

classes of clues

There are also other classes of clues like declarative-knowledge, word association and thematic clues as well as structural clues that use the number of letters in the

target word or refer to a specific letter in a specific position of the word [Nic11]. It is also possible to use these type of clues in educational crosswords although there often simpler clues and definitions might be used.

symmetry

Standard crosswords also provide 180 degree rotational symmetry so that if they are rotated, the location of the black and white cells remains the same. Furthermore, American-style crosswords fulfill “all-over interlock", which means that no part of the grid is fully delimited by black cells[Aml].

NPC

Very popular are the daily crosswords of the New York Times, which have a fixed size of 15 x 15 fields and are often themed. They are created by a constructor because humans are still better at constructing high quality crosswords puzzles than technical generators [Aml]. This shows how complex this NP-complete problem is [CA20].

Fortunately, for the use case of this thesis weaker conditions apply. Learners do not necessarily need perfect crosswords like the ones that New York offers. Also characteristics like rotational symmetry and ’all-over-interlock’ can be neglected, since the only strong requirement is to include all answer words. The main goal is to include a new task type with gamification and puzzle character to make the learning progress more enjoyable and efficient. The focus is, how they can be integrated appropriately into gamebooks not on the crossword puzzle itself.

3.1.2 Designing Guidelines

important factors

There are several design principles that influence the quality of a interface. Important is, that the user immediately understands the subject matter. The nature of the content should be reflected in a visually engaging and aesthetically successful way. Also a visualization of information helps the understanding of otherwise incomprehensible data. Furthermore, visual metaphors help to build associations and use previous knowledge. In general, also depending on the content type, user intention and interface type an obvious start, consistent logic and feedback are a few of the most important design principles [ABE08].

control

Here the the feeling of control of the user and the reduction of the user’s memory load as well as general consistency are crucial factors of UI efficiency and the perceived quality of the software for the user. Allowing the user to maintain control can be achieved by not forcing the user to execute unnecessary or undesired actions, offering flexible interaction, hiding technical details from the casual user and show results on the screen.

memory load

The reduction of the memory load can be implemented by using meaningful default options, using real world metaphors that are already known by the user and structuring the interface hierarchically so that most important content is displayed first while more details are revealed if the user shows more interest using for example a mouse click on a certain button.

consitency

Factors that improve the consistency of an UI are giving an overview where the user has come from and what alternatives he has, using the same design rules for a set of applications and try to meet the user expectations of how the interface should behave. Furthermore icons should be self-explanatory and actions easy to remember to simplify the learning of the basis operations for users [Sri14].

human-centered design

Additionally, the philosophy of human-centered design should be considered. Here the humans needs, capabilities and behavior are put first in the design process to create a positive user experience. Besides using visual signifiers that show where and how actions can be done, it is also possible to use constraints and forcing functions to avoid undesired usage and prevent user mistakes [Don13]. So for example buttons could be disabled if another action has to be done before using it.

When constructing learning material there are additional factors to consider during structural and content wise design. Besides the multimedia-principle, the spatial continuity principle, the segmentation principle and the signalizing principle for the design of context representation, the use of understandable and easy to distinguish items as well as the usage of colour coding can be very useful [RH21].

3.2 Technical Foundations

3.2.1 The DiGaBo Project

Angular, JS node.js, mongoDB

The DiGaBo Platform is an Angular project using TypeScript (stricter version of Java Script) with two front end projects: the editor and the UI. It uses a node.js backend and mongoDB as a database where the gamebooks are saved as JSON data. In the Angular project, there are components for every task type, for example Multiple Choice, Matching Pairs and now also Crossword [SN21].
The core part of the editor is the `EditgbComponent` that displays the canvas with the pages of the gamebook as well as the input options for the task types and the task component, while for the UI it is the `GamebookuiComponent` that loads the gamebook content.

3.2.2 Regular Expressions

use in implementation

The implementation uses regular expressions several times to check for specific characters. These characters help searching for placeable words in the crossword generation algorithm and saving additional information into the grid representation in the UI for the visualization of the puzzle. Therefore, regular expressions are briefly explained here.

search patterns

Regular expressions are used to find specific sequences of characters in strings (e.g. 'aa' in 'Vaals'). The Regular Expression describes this search pattern of the sequence. JavaScript has the object RegExp for regular expressions that is able to search, replace and validate the expression in a string with the predefined methods match(), replace() and test(). Additionally, there is for example the global flag /g to find all matches instead of only the first one. There are also the typical quantifiers *,? and {n} for 0 or more, zero or one more n occurrences of the character before it as well as assertions like ^ for the beginning of a string or $ as the end of a string. Patterns like [0-9] and [A-Z] are also possible to use a character of a character class [Wik, w3s].

RegExp

These are just a few examples of what is possible with regular expressions in JavaScript that are used later in the implementation of the crossword component.

3.2.3 Angular

Angular is the JavaScript framework that the platform DiGaBo uses. TypeScript is a more specific version of JavaScript that is applied to develop web platforms in Angular and therefore part of the front end.

structure

Angular uses components that represent pages or sections of a page and contain a HTML for the visualization, a TypeScript file for the logic and a SCSS file for the styling. Components can be integrated into each other and be related as parent and child, where parent refers to the super ordinate component. More complex logic is often outsourced into a service, e.g. the HTTP-service [angc].

Another principle of Angular is modularity so that the application consists of manageable and reusable parts [angf]. Furthermore, Routing is used to connect URLS to components for navigating through an application and displaying different components while enabling route guards and lazy loading to optimize performance [angb].

twp-way-binding

Morveover, Angular offers option like the two way binding that can be used to connect a value with an HTML element so that changes are updated simultaneously [angd]. `*ngIf` can also be used to render HTML elements only if a condition is true while `ngOnInit()` is in Angular the methode that is called after the input initialization for further initializations [ange, anga].

Chapter 4 Implementation

With the knowledge about important factors for efficiency, the platform DiGaBo and existing approaches for crossword generation (Section 2 Related Work) and the foundations of crossword creation, design guidelines and technical tools like RegExp (Section 3 Foundations) as well as the basics of Angular, the goal of crossword puzzles as a new task type for the platform DiGaBo can be reached. The main requirements are intuitivity, simplicity and the functionality to create, generate and evaluate crossword puzzles that include all answer words.

4.1 General Structure

Firstly, it is important to understand how the UI, editor and backend are connected. Therefore figure 4.1 illustrates the interaction between them. The crossword can be created in the editor by a teacher by giving clue and answer pairs as an input. When loading the preview or while saving the page if the preview function was not used before, the editor sends a HTTP request to the backend with the question answer pairs and further information. Then the crossword is generated in the backend where the logic of the algorithm can be found as a service. The created crossword with a description is returned to the editor in the HTTP response. Where it is displayed and offers the possibility to choose between different grids and to use a solution word. After this, the created grids get saved into the gamebook. So the crossword generation happens in the backend, the editor saves it into the gamebook and the backend saves all gamebooks into the database. This relieves the browser and outsources the logic to the backend.

input

req, res

generation

save grid

visualization

When a gamebook in the UI is loaded, the UI receives the saved crossword information and can load the visualization of the crossword and evaluate it when it was solved. Additionally, hints are available and visual feedback for the learner is given by colouring the cells.

4.2 The Editor

As mentioned before, crossword puzzles can be created in the DiGaBo Editor in a gamebook. In more detail, the educators create a new page and besides adding a title and more information, they can select crossword as a task type and the newly created `CrosswordComponent` appears below the task type buttons. Here, two `mat-form-fields` with label and input with placeholder are used, to collect the description/clue and answer word for the crossword and save them into an two dimensional array. This array starts with the value ["",""] so that the first pair can be edited

Crossword Component

input

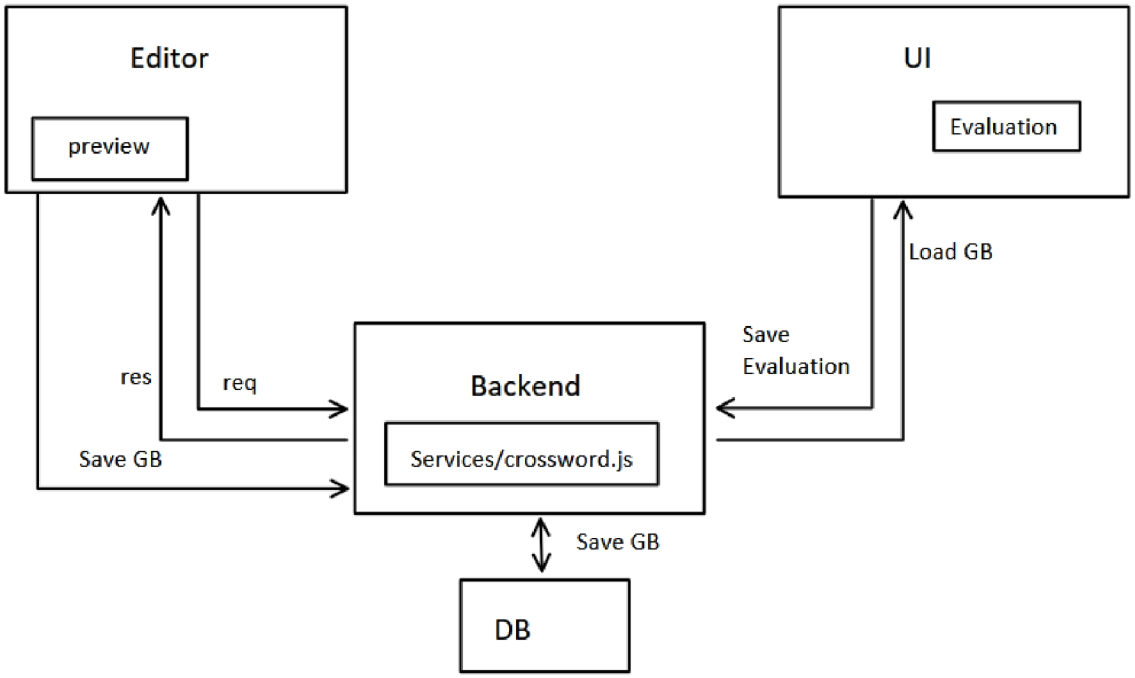

Figure 4.1: Sketch of the interaction between Editor and UI front end with Backend and Database

in the input field and is saved with the `[(ngModel)]` two way binding from Angular. So the question and answer array is displayed using `*ngFor` with an index i that allows deleting the i-th entry with the delete button next to the i-th pair using `.splice()`. Each of these pairs is surrounded by a border to visualize their connection. Furthermore, there is a button to add another clue answer pair.
The already in the project included translate service is included in all of this work added text in HTML to ensure that it can be displayed in different languages.
To save all the important information, the class `crosswordEntires` is created which can be seen in figure 4.2.

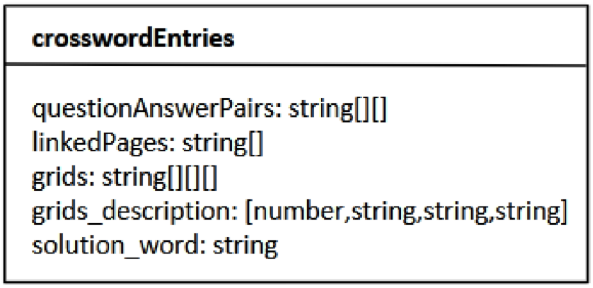

Figure 4.2: Custom data type for important information for the crossword generation in the editor and UI

But since it is hard to imagine for the teacher how the final crossword grid turns out, a preview pop up was added that is shown in figure 4.3. With a click on the green preview button the function `loadPreview()` opens the new `CrosswordPreviewComponent` as a dialog and hands over the collected crossword data. Since it has no close button except the save button, it can be closed with ESC or clicking outside of the pop up.

preview pop up

This function also saves the updated data, that was modified in the pop up.
The pop up is the component ih which the crossword generation in triggered. When it is open, the `NgOnInit()` function from Angular checkes if there are already grids saved that can be displayed. If this is not the case, up to 3 grids are generated in the `load()` function that sends a HTTP post request with the clue answer pairs to the backend where the grid and grid description is generated and saves the via HTTP response returned data. This first loading of the preview grid is important to have a visualization faster without clicking a button.

request grid generation

Additionally, it is important to consider that crossword puzzles only use capital letters. Therefore, the function also changes all input answer words to uppercase so that it is robust to the use of small letters. Furthermore, special characters as 'Ä', 'Ö', 'Ü' and 'ß' are replaced by 'AE','OE','UE' and 'SS' to enable more intersections.

There are multiple advantages when up to three grids are given in the editor. If a group of learners uses the digital gamebook together, all their grids contain the same words but the placements can be different. So every learner gets a randomly chosen one of them displayed and since they do not view the same they can not copy from each other and if the gamebook is played again the different layout leads to more variety. This ensures that the knowledge is learned not the grid itself. But the educator also has the possibility to give all learners the same crossword grid by deleting all except for one. This also offers the possibility to exclude non-aesthetic grids and to regenerate the grids again.

multiple grids

To realize this, the component has variables for the current grid Index in the array of up to three grids and the current grid as a string array. This is the grid that is currently displayed. To change between the generated grids, two buttons with arrows can be used. Those buttons also check if the grid is the first or the last and if that is the case disable the corresponding button to visualize the bound of the array.

As it can be seen, in the next section the current grid array does not only contain letter but also additional information in form of non-letter characters, for example the word numbers in the cells that include the first letter of the word. Therefore the grid is displayed using a self-written method called `extractLetter()` that matches the letter in the field using `RegExp` and returns it.

display grid

Crossword puzzle also often offer solution words that are compounded of letters in the grid. But the solution word is optional. Nevertheless, there are convincing reasons for the use of them. On the one hand, they are part of the gamification effect that gamebooks and the new task type support, since they offer an additional feeling of challenge and success when the word is found. On the other hand, they can also give hints for solving the crossword puzzle if the word can be guessed and therefore gives a letter in the grid as a hint for other words. This is why this implementation offers solution words but also gives the option to omit them by leaving the input field for the solution word empty.

solution word

This also requires another function that is called if the save button is pressed. First, the button only saved the changes and forwarded them to the crossword component where they were saved into the gamebook page. But now it also has to firstly check, if the solution word is possible to avoid unsolvable crosswords for the learners in the

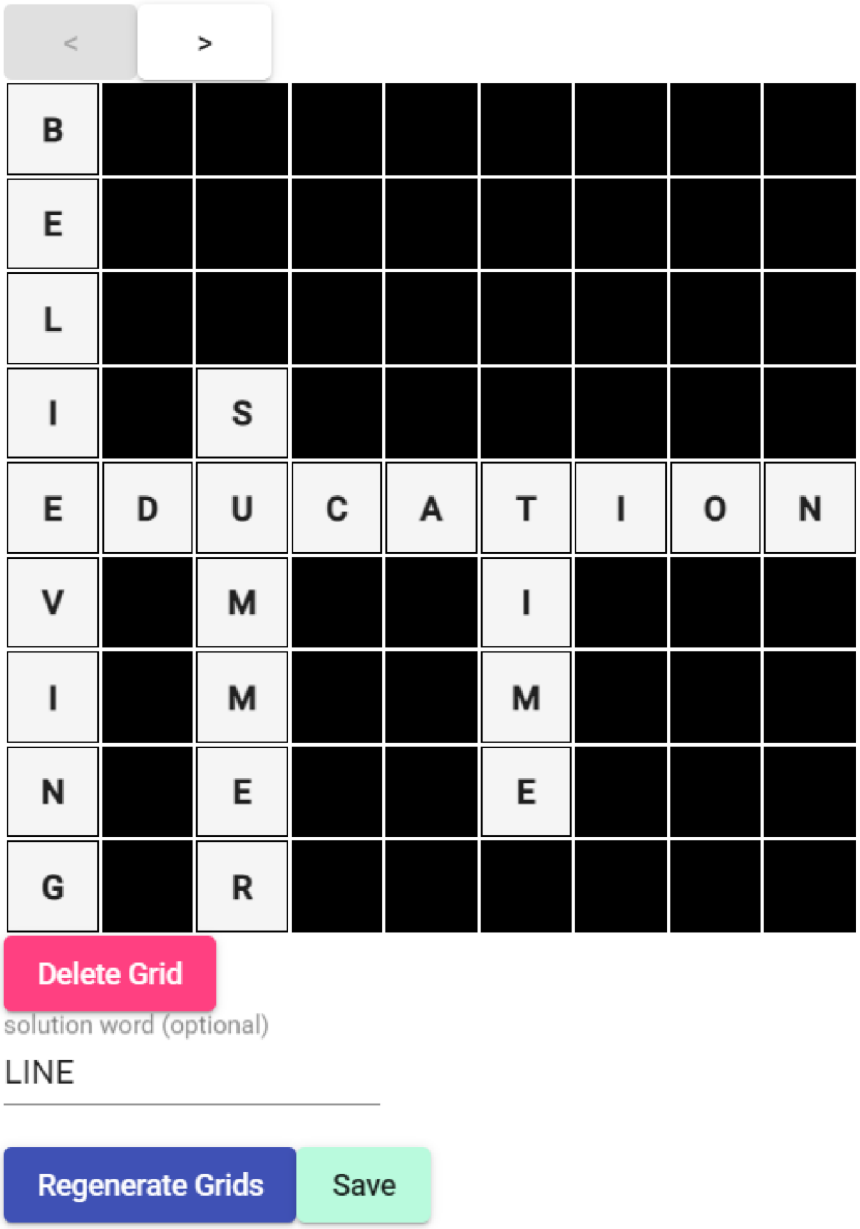

Figure 4.3: Screenshot of the implemented pop up with the preview of an example crossword using the solution word LINE, word definitions taken from an online dictionary [dic]

UI. Here, a solution word is possible, if for every of its letters there is a cell in the crossword grid with its letter. This is checked by creating a copy of the grid and after reducing it to fields that contain letters, every letter of the solution word is found and deleted in it. If the word is not possible, a red label next to the input field tells the user that the solution word is invalid.

So in the editor the clue and answer pair are given as an input, the crossword generation is requested and the result saved as well as displayed and a possible solution word can be added.

4.3 Crossword Generation

requirements

One of the core parts of this thesis is the crossword generation, so a fitting crossword algorithm had to be chosen. As mentioned before (Section 3.1 Foundations), in this use case, weaker conditions for the generated crosswords apply. The focus on the use of crosswords in educational gamebooks not on generating best-possible grids. The most important requirement is, that all answer words are contained in the grid which is not necessarily required in the algorithms presented in Section 2.4 of Related Work. Connectivity is desirable but all words have to be used even if for example a words has no intersections with other words.

influence of Joshi and Agarwell

Since no fixed gridsize is given, the scenario is very similar to the conditions for the algorithm of Joshi and Agarwel. The here implemented algorithm is inspired by it because they justified their algorithm convincing and proved its efficiency. (Section 2.4) Implementing it in JavaScript has the advantage that there is no need to integrate another programming language like Prolog into the project and because no library is used for the implementation, the code is more comprehensible as well as it can be easily modified and adjusted to several ideas.

Therefore the implemented solution of this thesis is strongly influenced by this algorithm even if a few changes are made because of the different conditions. The biggest changed condition is that every word has to be used in the grid. Therefore instead of using the iteration number as a termination condition, the word number is used to ensure this. Moreover, since all words have to be part of the grid, the use of randomization is added to provide different grids and another ranking criterion for the grids was chosen. The algorithm of Agarwel and Joshi is also only given in their paper as pseudo code and therefore the functions might be implemented differently but with the same goal in mind. But is also necessary to consider the well-documented performance of the inspiring algorithm does not apply for this new implementation since there are multiple adjustments that change the efficiency and performance.

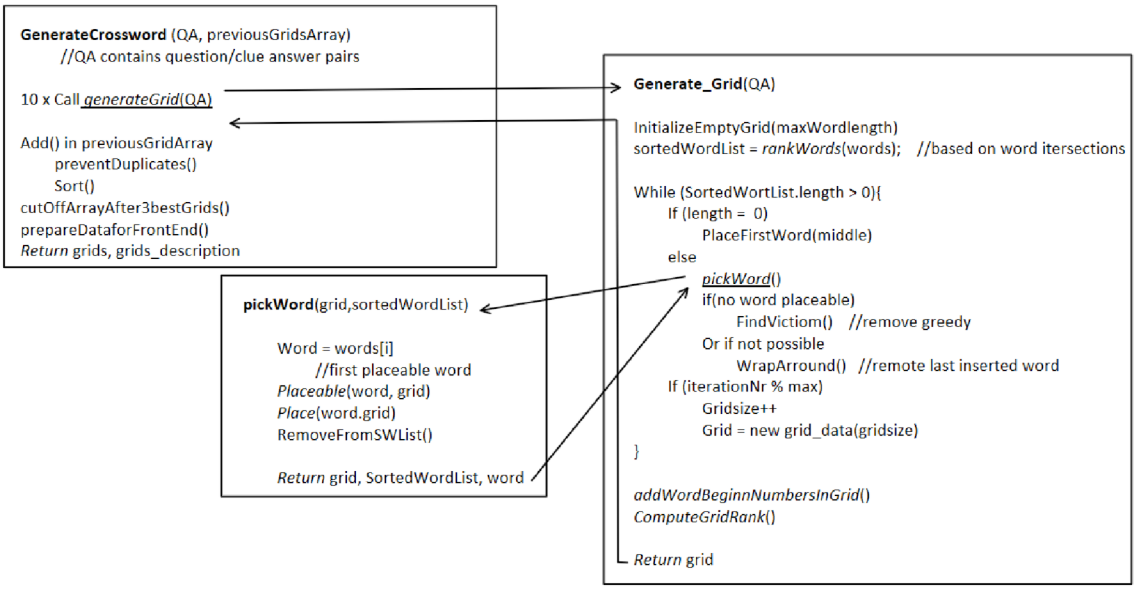

Figure 4.4: Structure of Crossword Generation Algorithm

structure of implementation

The resulting implementation of the crossword generation consists of a function called `generateCrossword()` as show in Figure 4.4 that calls the function `generateGrid()` ten times so that more than the three displayed grids are generated. Then the array of grids is cut off after the three best grids and they are saved for the editor front-end. For generating the grids the words are ranked based on word intersections. To not favour the first words with the same number of intersections every grid generation the sequence of clue and answer pairs is randomly rearrange, which also extends the diversity of generated grids. To avoid blocking intersections, words that have no intersections with other words are added at the end of this list.
The placed words are added within multiple iterations that include picking the first placeable word and place it and using a greedy removal strategy or removing the last word if no word is placeable. After a certain number of iterations the grid size is increased and the grid reset to find a better solution. This number is currently twice as many as answer words but it is possible to experiment with other values.

data representation

This implementation uses three custom classes as shown in figure 4.5 to save all relevant data for the grid. The custom data type `grid_data` represents generated grids by saving its word, the size and its rank. The grid size does the refer to the real grid size but its number of rows.The grid it self is an array of element of the class `grid_cell`. Furthermore, it has a function called `getGridArrayfromCellArray()` which turns this array into a String array that contains the characters of the grid cells.

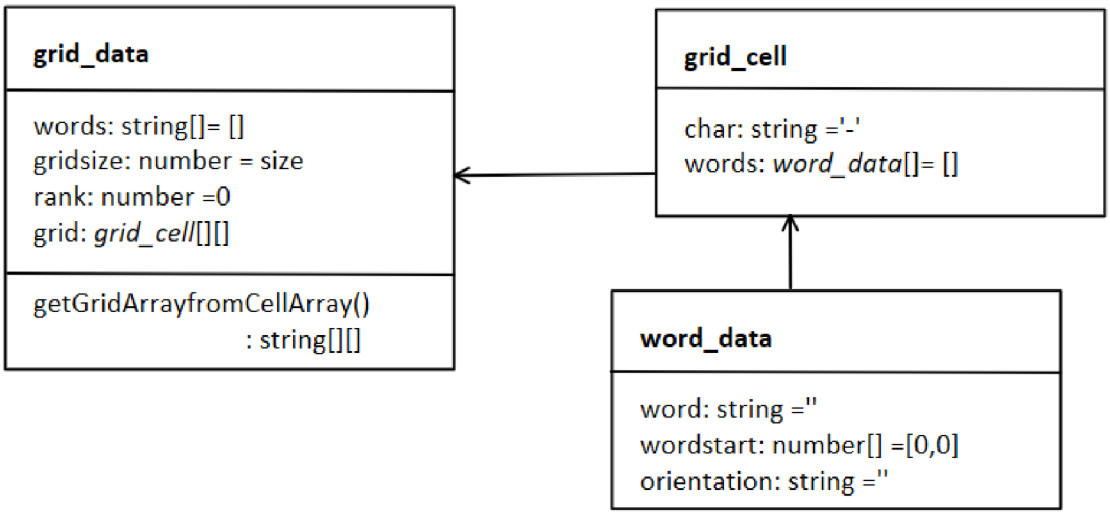

Figure 4.5: Custom Data Types for Crossword Generation in Backend Service

`grid_cell` saves a the characters of a cell and the words the cell belongs to as a `word_data` array. For the words the custom class `word_data` is used which saves the coordinates of the cell in which the word starts and its orientation. Since Java Script does not natively support Enumerations, the orientation is a string where 'H' is used for horizontal and 'V' for vertical. Later in functions like `place()`, 'B' is also a valid value and indicates that both directions are possible.

sort words

As Joshi and Agarwel found out, it is most effective to add words following the pebbles and sand principle and therefore ordering them ascending by their intersections on word level [CA20]. So to compute this sorted word list, where the first element is the word that should be added next, all combinations of two words are considered and for every word the number of different words that overlap with it is calculated. With this rank vector the word with minimal intersections ($\neq 0$) is added to the `sortedWordList` until all words (also words with =0) are in it. During this procedure the maximal length of the words is also calculated because it will decide the first used gridsize.

placeable()

RegExp

Very important for placing words in the grid is the function `placeable()` that checks if a word is placeable and then returns the start position of a good placement with the orientation.
To search for a fitting gap in the grid, the grid as a string array is transformed into a string using '$' for the start of the grid and the end of lines. Then a RegExp pattern (further explained in Section 3.2 Technical Foundations) for the word is created. Since empty cells are represented by "-" which later represents the black cells, this character is also part of the regular expression. So every word starts and ends with [$-]. Between that, it uses [C-] for every letter C of the word (e.g. [$-][B-][I-][T-][$-] for 'bit'). The [$-] is necessary to avoid the overlapping of two words with the same direction so that for example it does not appear that the two last letters of one word are the two first letters of another words and the visual distinction is eased.
Then using RegExp and the function `.matchAll()` an array of all matches is created.

Figure 4.6: Demonstration of the problem of letters that seem visually to belong to other words

Since this only finds horizontal matches, the grid is also transposed and a new string created to create another array representing the vertical matches.

Furthermore, all matches are checked again if they really fit and there is no letter in the field before or after the word that is not part of it because it might visually appear like that. This would make it hard to distinguish between words as for example with the 'E' in figure 4.6. It would also be possible to use word length number at the end of clues like used in British crosswords (Section 3.1) so that the learner can count which cells do belong to the word but this solution makes the distinction between word gaps more intuitive and faster to understand.

Additional Check

This additional check is done by checking for each cell of the match if in the two orthogonal neighbour cells there is no word with the opposite orientation that does not overlap with the placeable word. So there is no word ending of that word next to the match.

Next the match with the most intersections with other words is used to calculate the position in the grid from the index of the match using the formula $\left\lfloor \frac{index}{gridsize+1}, index \mod (gridsize+1) \right\rfloor$ where if the match is vertical the coordinates need to be swapped. So the function can return this position, the orientation and a boolean to decide if the word is placeable.

With the intention of not favouring horizontal over vertical matches it is randomized which array is checked first and if the best match in both are equal, an arbitrary one is chosen.

When a word is placeable the function `place()` is called to integrate it into the grid. If the orientation is 'B' vertical and horizontal is possible and one of them is chosen randomly. The first word of the grid is thereby placed in the center of the grid with the start position $(\left\lfloor \frac{gridsize}{2} \right\rfloor, \left\lfloor \frac{gridsize-word.length}{2} \right\rfloor)$. Then the letters are added into the cells following the given direction and in every cell the `word_data` including the start position, orientation and the word itself is added. Additionally the word as a string is added into the array of words in the grid of the `grid_data` element. After every placed word, the sorted word list is reranked again by the number of intersections with the words in the grid but words without intersections are added last.

place()

Both presented functions `placeable()` and `place()` are called by the function `pickWord()` that is responsible to chose a placeable word, place it and remove it from the `sortedWordList` as well as returning it.

pickWord()

But if no word is placeable, an empty word is returned and a removal strategy is required to make any words placeable without immediately increasing the grid.

removal strategies()

Joshi and Agarwal propose using a greed removal strategy and if no word to remove can be chosen with this method they remove the last word that was added[CA20]. This is also what has been implemented in the functions `findVictim()` (greedy removal strategy) and `wrapAround()` (remove last inserted word). The greedy removal strategy here removes the word where the most words are placeable if it was removed.

Here a helper function is used for only removing letters that do not belong to other words. As proposed in the paper, to avoid loops due to the removal, the priority of removed words is decreased. This is done here by adding them at the end of the `sortedWordList`.

Using the presented functions and strategies, `generateGrid()` generates a grid and after that the words in the grid are enumerated according to their order they were added into the grid. These numbers are also added in the string of the cell that represents the words first letter to show the field with a word beginning in the UI for the learner. When the grid does not contain any intersections, the next grid generation starts with a bigger gridsize to offer more possible gaps for words.

grid ranking

In addition, after the generation of a grid the score of it is calculated as a criterion for comparison. There is no clearly defined criterion to rank grids and to describe their quality but there are a few approaches. One of them is the grid utilization which is the percentage of white and black cells focusing on the fullness of the grid. The algorithm of Agarwal and Joshi for example the average of white cells is 45-55 %. [CA20]. The number of words instead of letters can also be an option for comparing generated grids. Furthermore characteristics like connectivity or symmetry could be possible.
Since the number of words, that is often used, is fixed in this use case and the percentage of white cells has often the same value for multiple grids, this implementation uses the number of intersections to focus on connectivity that is an important goal for the visual representation. But it would also be possible to use more complex criteria like choosing the lowest maximal connected area of black cells to focus on more aesthetic features of crossword puzzles. Or to combine the number of intersections with the fullness of the grid with the goal function $intersections \cdot \frac{letters}{width \cdot height}$ [JD22].

generate-Crossword()

With this ranking it is possible to add the newly generated grids in an array containing `grid_data` elements and rank them based on this value in the function `generateCrossword()`. Duplicates of grids are not added into the array and since only the three best grids are displayed in the editor. Therefore, the array of grids is cut off after the three grids with the best grid ranking. Before returning the data to the editor, the grids are transformed into a string array to simplify visualization in the front-end and an array `grids_description` is created. This contains for every word its index, clue, the word itself and its orientation because this is needed to display the clues for the learners.

randomiza-tion

Some elements of this implementation use randomization. Besides the reason of not favoring horizontal over vertical matches in the function `placeable()`, this strategy also helps to generate different and maybe even better grids when the user presses the button for regenerating grids in the editor because without this, the results would always be the same since all words have to be part of the resulting grid. So randomization is used a few times in this implementation but could also be used more frequently to improve the algorithm further.

4.4 The User Interface for learners

UI

When a gamebook using the crossword task type has been created, it can be played in the User Interface of DiGaBo. So when the page with the crossword puzzle is opened the fillable grid and the clues for the answer words are displayed as shown in the figure 4.7. Furthermore, there are buttons for clearing the grid, receiving a hint and for the evaluation and feedback. Here it is especially important to provide a simple

and intuitive surface because the UI is more often used for playing the gamebook than the editor where the it is once created and a few times edited. Often the learner also does not know the platform as well as the teacher that might use them more frequently, so even if the learner plays a digital gamebook for the first time, the UI should be easy to understand.

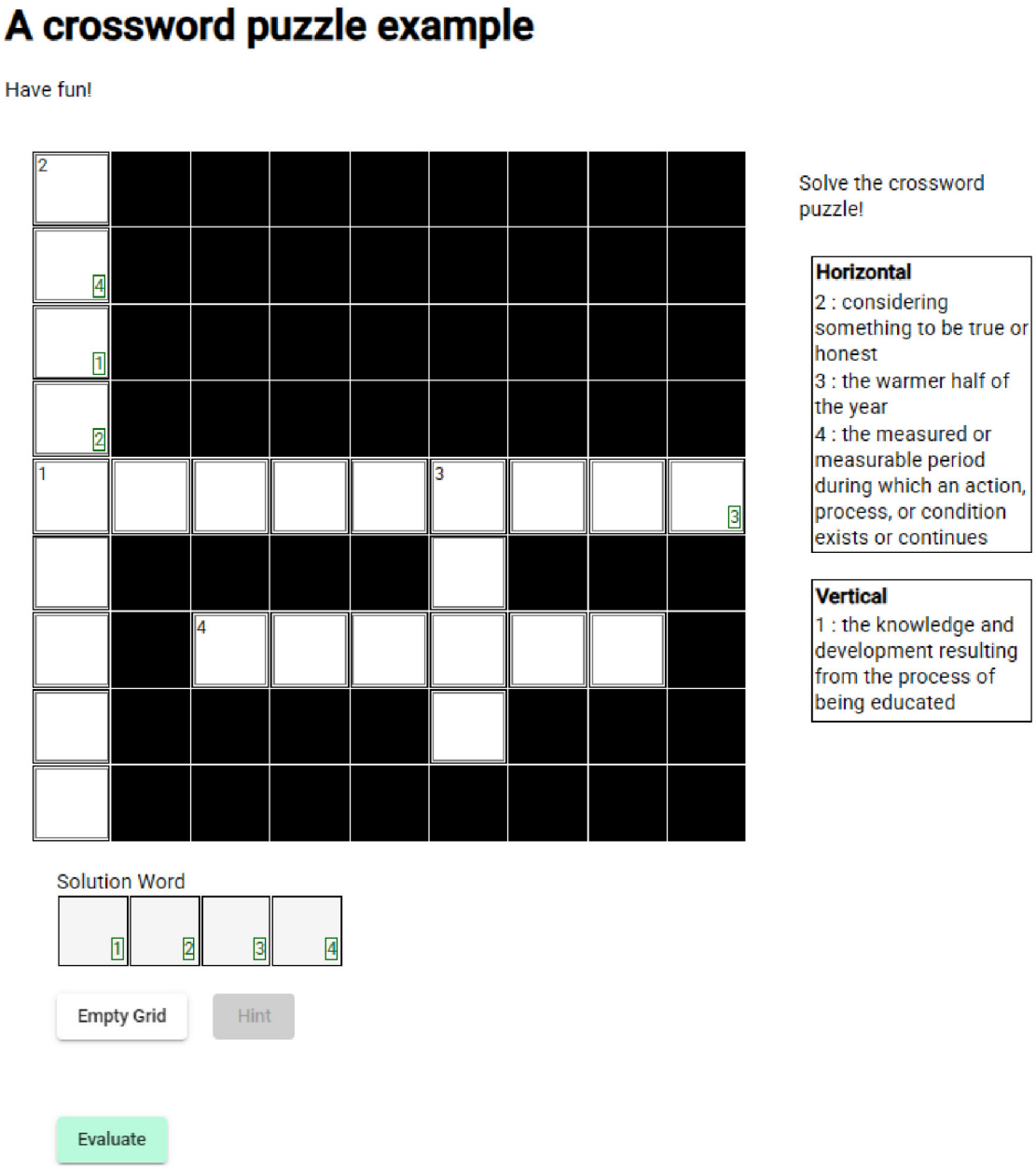

Figure 4.7: Screenshot of example crossword in the UI

displaying the grid

As it can be seen in figure 4.7, the grid is displayed with white and black cells. This visualization format was chosen instead of the gridless version 2.5 because it is less associated with education and therefore causes more interest [EKC83]. Small numbers in the top left corner of the cell mark word beginnings, while the green

bordered small number in the bottom right corner highlight the letters that are part of the solution word. This is realized by displaying the array `currentGrid` from the new `CrosswordComponent` using a table with `*ngFor` and `[ngModel]`. This variable is initialized with the solution grid which is emptied by the function `clearGrid()` that replaces all letters as well as the characters '=' and '?' (important only for visual feedback later in this section) so that further information like the word begin numbers and the black cells remains in the array. In addition, the function also changes the label text to an invitation to solve the crossword puzzle and enables the later explained editmode. For displaying the grid cells, the function `TrackBy` is used to identify cells distinctly using the index because otherwise cells would be mix up resulting in unintended letters in the visual grid and counterintuitive behavior.

clearGrid()

The main difference compared to the table in the editor is that the ngModel two-way binding that angular offers is splitted in its both directions to separate the displayed letter and the other information in the array fields. Additionally, it allows disabling editing in the table as well as an input check so that only letters can be entered. The extracted letters of all cells are displayed so that the entered letters are shown as well as black cells using CSS for ngClasses as in the editor. The inserted input in a cell is given via `(ngModelChange)` to the function `onCellInput()` that deletes old letters from the cell and adds the new letter to the other characters that might be saved there before. To Avoid cheating like entering all letters into all cells, which would by considered as correct by the proposed solution, the input field has a size limit so that only one letter per cell can be entered.

data representation

To use the information of the other characters in the string of a cell beside letters there are a few other functions. So this string can contain one letter, a number that represents the word beginning of the corresponding word, a number framed in '!' to mark the cell as the corresponding letter of the solution word and the characters '=' and '?' that are only relevant for the evaluation and the visual feedback. So the functions `isWordStart()` and `isPartOfSolution()` return true if the corresponding characters are part of the cell while the functions `extractLetter()`, `extractNr()` and `extractSolutionNr()` return their values.
The solution word is saved in the UI during the `NgOnInit()` method as the coordinates of the corresponding cells that contain the letter so in the UI their value is simply displayed separately below the grid and filled in when the letter was filled in above. Additionally the number of the solution word letters are added into the string of the corresponding cells with a '!' before and after the letter

hint button

Since it is important that tasks match the skill level of the learner, the crossword puzzle has to be challenging but not to hard to solve to avoid frustration. So to help the learner, besides the option to leave fields empty when clicking evaluation and perceiving feedback as well as the solutions, a help option was implemented. Now the learner has to option to click the 'Hint' button which is enabled using a timer after 30 seconds to reveal a random unfilled white cell. The time constraint of 30 seconds was chosen to force the learners to first think on their own and not filling the whole grid by only using this button. After the first hint is used, the button enables itself every 15 seconds to reduce the needed patience of the learner. These values are not

tested on a sufficient number of people, therefore, it could be useful to change the time intervals in the future or to individualize them even further depending on the previous knowledge and skill of the learner to adjust difficulty. Furthermore, it would also be possible to enable only three hints per puzzle to speed up the time learners might spend with this task. But since an advantage of gamebooks is that users can choose their own speed and level of help, this was not implemented in this solution.

evaluation

When the crossword grid is filled as much as possible, the learner presses the evaluation button, which triggers the most important part of the UI besides the visualization of the grid. Here the previous mentioned editmode is disabled and turns the input fields into read-only field. Additionally, the hint and evaluate button are disabled while the clear button can be used to repeat the task.

succes

Since a crossword puzzle has a rather complex solution compared to other task types like singe choice, it can be discussed when the grid is defined as successfully solved. There are many possible approaches and there is no clear preference. The learner could be expected to fill all white cells right, but this requires the knowledge of all answer words without making spelling mistakes. But depending on the word number, it can feel counterintuitive to label the solution as wrong if the person filled in 90 % of the answer words or cells correctly. To keep the motivation of the learner up, this implementation does not expect a perfect solution and labels a solution as successful if 66% of the cells are filled correctly. For the purpose of a simple and comprehensive Evaluation, the percentage refers to cells instead of word (slightly favouring longer words) and does not give greater weight to cells that are part of the solution word. Another reason for not giving the solution word more value is that it is only part of gamification and might give the learner another hint but it is does not influence the learning objective and does therefore not represent the knowledge that is represented by the words in the grid.

xAPI

Besides the evaluation itself, the button also triggers which page is opened after this task depending on the success of the solution and the xAPI statement that in added to the local storage like in the other task types for later evaluations and analysis of the gamebook play through. Here the success, the percentage of right cells, the number of hints needed and the next pages are contained.

grid navigation with arrow keys

The usability can enormously improved by enabling to fill the grid using the arrow key since this is be more intuitive than clicking on all input cells. Additionally, it is the faster option. For this template variables for the input cells are used, so that the HTML elements can get accessed with `@ViewChildren` in the TypeScript code. Since Angular does not order them as in the table that represents the grid, custom data attributes for row and column are used to access the cells and to sort them ascending. So in the function `ngAfterViewInit()` the input references are sorted and the focus starts in the first cell of the grid. The focus is always on the cell that is filled and can now be changed using the arrow keys since the function `handleKey` handles the keyboard events and changes the active column and row that is focused. Since it takes a few seconds until all inputs are rendered by Angular, the function `ChangeDetectorRef.detectChanges()` does local change detection checks [cha] and therefore, the arrow keys, can be used without waiting time.

For the learning progress it is essential to obtain instant and comprehensive feedback [MT13]. Therefore, visual feedback is given, when the evaluation button is clicked as shown in figure 4.8. Correct cells are displayed green while incorrect cells are filled orange. In the table that displays the clues for the searched words, the answer words are also shown because that is where they belong to in terms of content. Additionally, the label text tells the learner the reached percentage and has predefined text for 100%, more than 66 % and solutions that are not successful, that encourage the learner and reduce the effort to integrate the crossword puzzle into the gamebook for the educator since this can be used as a simple connecting part to the next task or information block.

visual feedback

The visualization of the correct and incorrect cells is implemented in a similar way to

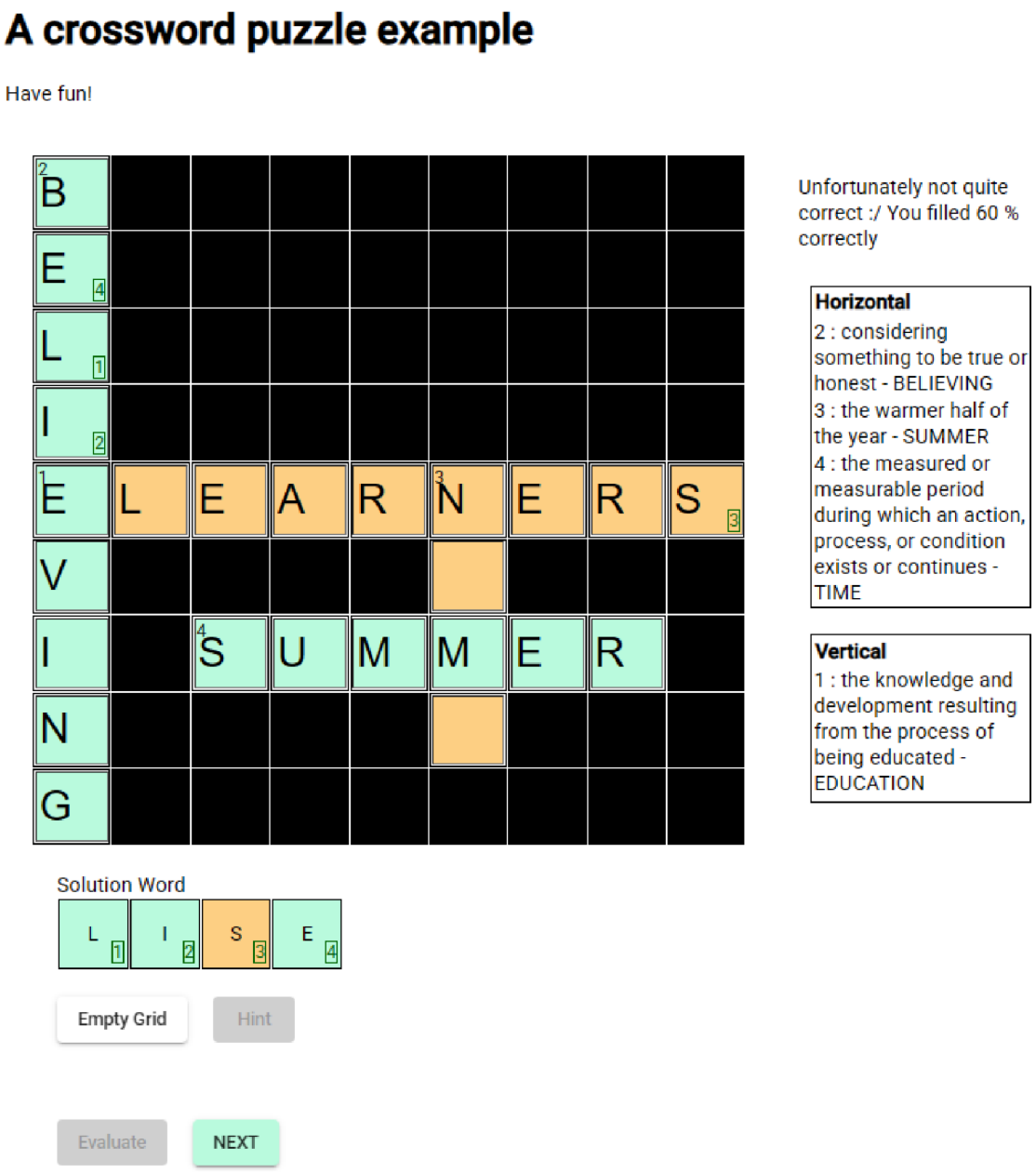

Figure 4.8: Screenshot of the visual feedback for an example crossword puzzle

the black cells in the grid or the word begining numbers. In the HTML tag `ngClass` is used to give assign them to a class depending on which method of `isWrong()` and `isRight()` returns true. So if the cell contains '=', `isRight()` returns true and the cell with now the class `right` is coloured green by CSS. In contrast, '?' represents an incorrect cell and is colored red. The solution word cells below the grid are colored similarly.
The 'continue' button only appears after the grid was evaluated so that the task can not be skipped.

Chapter 5 Conclusion

problem definiton

The goal of this thesis is to integrate the functionality of a crossword puzzle as a new task type into the platform DiGaBo.
Therefore, a concept for the given problem definition has been developed and implemented into the project. This includes the creation of a new component in the editor and UI, a preview component in the editor and a crossword generation service in the backend.
The main parts of the feature are the possibility to give clues and answers into the editor website as an input, to generate a crossword puzzle and evaluate it in the backend and to solve it in the UI website. Furthermore, a preview, a solution word, a help option and visual feedback were added during the process.

goals

The feature as a whole focused on the following goals and relevant aspects from related work:

- simplifying the use of crosswords for educators
- intuitive and simple interface
- improving the learning progress
- increasing the joy and motivation of learning

These are only the goals that are relevant for the implementation since the role of clues is also crucial for the success of learning with crossword puzzles. But this is the responsibility of the gamebook creator.

easy to use

The first aspect refers to the effort to create a crossword puzzle which is for a normal person a complex task especially when only words should be included that are relevant to the topic. So instead of giving an empty input grid where the gamebook creator has complete freedom in choosing and placing words, the task for them is reduced to only think of clue and answer pairs. This relieves the process so that the danger of using filler words just because they fit into a gap is avoided. Therefore, the proposed solution reaches the goal to provide easy usage of crossword puzzles as part of a gamebook even if the educator has less influence of the resulting grid. Furthermore, no technical knowledge or programming skills are required to use the crossword generating function, because the technical details of the implementation are not visible to the normal user. Additionally, there are no word restrictions beside the assumption that answer words are unique and the the grid is always square-shaped. The editor also helps the educator by automatically turning the answer

words into capital letters and replacing special characters like ’Ä’ but nevertheless, it is also possible to enter other characters as ’;’ to create unsolvable puzzles since these cannot be entered in the UI. Though, since only educators use the editor with the intention of creating usable grids, entering these is counterintuitive and should not happen.
But the complex task of finding clue and answer pairs is still the task of the user and there are no instructions or guidelines provided on how to give useful clues and select effective answer words for different learning objectives.

intuitivity and simplicity

Besides the functionality, the interface should be easy to use as well. Therefore, the goal of an intuitive and simple interface is reached by offering not too many buttons and functions and giving the interface a clear visual structure. So colours are used to highlight important buttons, visual boxes connect the clue answer pairs and the preview is moved into a pop up.

improving the lerning process

Another goal of the new feature is improving the learning process. This goal can be measure by aspects that were further explained in the Related Work Section 2. Since the learner has to fill in the grid, the task is part of active learning and offers self-determination since the learner can decide how much time to spend with the task before evaluating the solution. It is also possible to decide how many hints to use and how many cells to leave empty for the evaluation.
Also the testing effect and the value of instant feedback is given, because the learner can always click the evaluate button and receive colourful feedback, the reached score as well as the correct solutions.

enjoyment

One of the most important goals of the implemented task type is the enjoyable learning process since the gamification is one of the main aspects of crossword puzzles. So this gamification aspect is reinforced by the feedback view that gives the percentage of right letter and colours the wrong and right letters as colourful visual feedback as well as motivating comments depending on the reached success. To avoid frustration the hint function was implemented and the grid ranking criteria is the number of intersections to maximize their number because intersections give additional information for other words.

fazit

So the proposed implementation follows the goals of simplifying the use of crosswords for educators, providing an intuitive and simple interface, improving the leaning progress and making the learning process more enjoyable. Even if further improvements like offering more instructions and guidelines for the gamebook creators and including even more gamification elements, the implementation offers multiple advantages and is therefore a good solution for the given problem.

Chapter 6 Future Work

clue generation

Since the proposed solution generates crossword puzzles from given input containing the clues and answers, the educators are responsible to choose them didactically appropriate as mentioned above. Therefore, it would be reasonable to offer instructions and training if needed. Another option that could be investigated further, is to also generate the clues using dictionary databases or Large Language Models (LLMs). For the use of LLMs and natural language processing, there are already a few approaches to realize automatic clue generation [AZ24, KZ23b, KZ23a, AP23]. This improvement would speed up the crossword creation process because the previous given information material could be used and less creativity is needed.

choice of solution word

But besides automation of clue generation, there are other possible improvements of the proposed solution that are not covered yet.
In the editor the solution word is currently typed into an input field and it is checked that the letters occur in the grid. But to give the educators more freedom to make certain words more valuable it would also be possible to make the cells of the solution word choosable by clicking on them. Another option to reduce the effort of the educator would be to give suggest solution words that can be formed by the letters in the grid so that one can simply be chosen by clicking on it. This could be realized by using a word database and would speed up the process of creating a crossword. Moreover, it would be useful to filter the answer words so that the solution word is a different word.

auto save

Furthermore, with the numerous pages that a gamebook might have and the additional pop up in the preview, it is now even easier to miss a save button because the user has to save the grid in the preview, the gamebook page and the book itself. Therefore, it could be very useful to implement automatic saving.

performance of generation algorithm

Additionally, the crossword generation algorithm can be further improved since the criteria for performance are not examined within this work because it focuses more on the use of crosswords than the quality of them. Other approaches to generate the crossword grids from the Related Work section 2.4 could also be used or integrated into the current implementation. Furthermore, not all edge and corner cases might be covered since it is assumed that the task type is used as intended.
Especially a good look ahead strategy, more use of randomization and a more complex removal strategy for the `findVictim/()` function might lead to better results. To give the user more freedom in the editor, the crossword generation could also considee previous generated grids. Then it would be possible for the educator to generate only on new grid keeping the other grids instead of regenerating new grids without using

the earlier created grids. But it should always be considered that the generation should not take too much time since the educator should not need to wait for the preview too long.

more randomization

Another approach for the usage of more randomization would be to choose the best match in the grid in `placeable()` random if multiple matches are equally good. An alternative option is to choose the match random with a weight function that gices matches with more intersections a higher chance.

In the UI there are also elements that could be added without endangering the goal of simplicity. There is evidence that time pressure can increase the focus and attention and improve the cognitive performance[IDR22, ST21]. Therefore, it could be useful to include a visual timer that tracks the needed time to solve the puzzle or as a time limit. Here the ngx-countdown could be used but since it is hard to apply the right amount of time pressure to avoid anxiety and frustration, this solution does not cover this idea currently.

time pressure

Also the definition of a successful solution in the UI is not necessarily the best one as mention in Section 4.4 (Implementation UI). Therefore, it can still be discussed and adjusted. While currently over 66 % of correctly filled letters are considered successful, it could also depend on the number of correctly filled words or consider a word as right if only one letter is wrong because it seems like a typing mistake. Without a study, there is no clear preference so it should be experimented with this definition. Additionally, if a timer is used, the needed time could also be taken into account.

defintion of right solution

Moreover, from this data it is also possible to compute a score on the feedback page and to show a small leader board there to increase the gamification part in the feature. But since a score board might only motivate strong learners and frustrate weaker learners, this option should be discussed carefully.

score board

Besides the evaluation of the grid, the evaluation data of xAPI could be analyzed to find factors that help to adjust the task type. Adding a time tracking during solving the task could help to measure how difficult the task is for the specific learner and give the educator feedback if one specific puzzle is too hard to solve.

analysis of use data

With more analyzing the data it could also be possible to adjust the hint option to the individual level of the learner. The level could be constructed by scores in and time usage of previous tasks so that the time intervals between the next hints could be adjusted. Another option to individualize the interval time of the hints could be the time period since the last key was pressed so the learner does not think too long about the answer and wastes too much time.

individualize hint option

Another simpler way to reduce the frustration and help the learner to fill the grid might be a link to another page with learning material that is needed to find the answer if the learner does not have the required knowledge. To give more motivating feedback, it would also be possible to use sound feedback that adds more stimulation.

Important is also more research on crucial factors for the success of educational crosswords and other criteria that increase the desired effects and make the use of

more research on factors

crosswords comparable. The usability of this implementation could be examined to decide if the User Interface is intuitive enough or if further information and instructions in the editor or UI are needed for example in form of small tutorial. Also a closer study on the used algorithm could help to improve it further and to decide which of the proposed improvements offers the best opportunities.

Appendix A Bibliography

[AA24] L. Baitleuova A. Aliakbarova. Active learning: Using crossword and word-search puzzles for teaching vocabulary to students of the medical faculty. *Eurasian Journal of Philology: Science & Education*, 195(3), 2024.

[ABA24] D. Andreev A. Bosakova-Ardenska. Design and Implementation of Educational Game Using Crossword Principles. *Engineering Proceedings*, 70(1), 2024.

[ABE08] M. Zender A. Blair-Early. User Interface Design Principles for Interaction Design. *Design Issues*, 24(3):85–107, 2008.

[Aml] D. Amlen. NYTimes: How to solve a crossword puzzle. https://www.nytimes.com/article/how-to-solve-a-crossword-puzzle.html [last access: 2025-08-15].

[anga] Angular: NgIf. https://angular.dev/api/common/NgIf [last access: 2025-08-31].

[angb] Angular Routing. https://medium.com/@schaman762/angular-routing-best-practices-and-advanced-techniques-2da3f47226f1 [last access: 2025-08-20].

[angc] Angular Tutorial Structure. https://blog.it-frankfurt.com/posts/2-angular-tutorial-structure/ [last access: 2025-08-20].

[angd] Angular: Two Way Binding. https://angular.dev/guide/templates/two-way-binding [last access: 2025-08-31].

[ange] Purpose of the ngOnInit method in Angular. https://www.geeksforgeeks.org/angular-js/purpose-of-the-ngoninit-method-in-angular [last access: 2025-08-31].

[angf] Understanding Angular Modules. https://codewithpawan.medium.com/understanding-angular-modules-a-comprehensive-guide-180100403396 [last access: 2025-08-20].

[AP23] G. Muley A. Nerurkar A. Phalak, S. Yevale. Automatic Question Generation. In Abhishek Bhattacharya, Soumi Dutta, Paramartha Dutta, and Vincenzo Piuri, editors, *Innovations in Data Analytics*, pages 235–244, Singapore, 2023. Springer Nature Singapore.

[AR19] T. M. Gureckis M. Bretzke F. Xu A. Ruggeri, D. B. Markant. Memory enhancements from active control of learning emerge across development. *Cognition*, 186:82–94, 2019.

[AZ24] S. S. Kadali M. Maggini M. Gori L. Rigutini A. Zugarini, K. Zeinalipour. ClueInstruct: Text-Based Clue Generation for Educational Crossword Puzzles, 2024.

[BO22] A. Ruggeri B. Opitz. Vorteile Durch Aktives Lernen Bei Schülerinnen Und Schülern, Vorteile Durch Aktives Lernen Bei Schülerinnen Und Schülern, 2022.

[CA20] R. K. Joshi C. Agarwal. Automation Strategies for Unconstrained Crossword Puzzle Generation. *CoRR*, abs/2007.04663, 2020.

[cha] Angular: ChangeDetectorRef. https://angular.dev/api/core/ChangeDetectorRef [last access: 2025-09-05].

[DB16] T. M. Gureckis F. Xu D. B.Markant, A. Ruggeri. Enhanced Memory as a Common Effect of Active Learning. *Mind, Brain, and Education*, 10(3):142–152, 2016.

[dic] Merriam-Webster. https://www.merriam-webster.com/ [last access: 2025-08-31].

[Don13] A. N. Donald. *The Design of Everyday Things*. MIT Press, 2013.

[EKC83] S. M. Crossman E. K. Crossman. The Crossword Puzzle as a Teaching Tool. *Teaching of Psychology*, 10(2):98–99, 1983.

[Fid] R. D. Purdiasih Fidrayani. Meta-analysis of the effect of crossword puzzle media to learning and students' social science learning outcome in elementary school. 8.

[GM97] P. Gray G. Meehan. Constructing crossword grids: Use of heuristics vs constraints. *Proceedings of Expert Systems*, 97:159–174, 1997.

[IDR22] C. A. Hutcherson I. D. Roberts, Yi Yang Teoh. Time to Pay Attention? Information Search Explains Amplified Framing Effects Under Time Pressure. *Psychological Science*, 33(1):90–104, 2022. PMID: 34860637.

[IZ22] J. Bushati I. Zadeja. Gamification and serious games methodologies in education. In *International Symposium on Graphic Engineering and Design*, pages 599–605, 2022.

[JD22] W. Wojna J. Dakowski, P. Jaworski. Quick generation of crosswords using concatenation. In *2022 IEEE Conference on Games (CoG)*, pages 590–593, 2022.

[Kub23] R. Kubala. The Aesthetics of Crossword Puzzles. *The British Journal of Aesthetics*, 63(3):381–394, 2023.

[KZ23a] G. Angelini L. Rigutini A. Maggini M. Gori K. Zeinalipour, T. Iaquinta. Building Bridges of Knowledge: Innovating Education with Automated Crossword Generation. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 1228–1236, 2023.

[KZ23b] M. Maggini M. Gori K. Zeinalipour, M. Saad. ArabIcros: AI-Powered Arabic Crossword Puzzle Generation for Educational Applications. In *Proceedings of ArabicNLP 2023*, page 288–301. Association for Computational Linguistics, 2023.

[Lem14] M. Lemke. Wie lernwirksam sind Online-Tutorials? Lernerfolgskontrolle und Evaluation bibliothekarischer E-Learningangebote. *Perspektive Bibliothek*, 3(1):59–84, Mai 2014.

[Let12] K. Letrud. A rebuttal of NTL Institute's learning pyramid. *Education*, 133:117–124, 01 2012.

[lex] Lexikon der Psychologie: Testungseffekt. https://dorsch.hogrefe.com /stichwort/testungseffekt [last access: 2025-08-15].

[LG19] S. Robra-Bissantz L. Grogorick, R. Finster. *Digitales Lernen fesselnd gestalten: Motivation beim Lösen verschiedener Aufgabentypen*. TUDpress, 2019.

[LP19] H. Treiblmaier L. Putz. Increasing knowledge retention through gamified workshops: Findings from a longitudinal study and identification of moderating variables. In *Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)*, 2019.

[LR08] M. Maggini-M. Gori L. Rigutini, M. Diligenti. A Fully Automatic Crossword Generator. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 362–367, 2008.

[McK] D. McKie. History of Crosswords. https://www.theguardian.com/crosswords/2013/dec/20/100-years-crosswords-first-new-york [last access: 2025-08-19].

[Mer16] W. Merkel. The Potential of Crossword Puzzles in Aiding English Language Learners, journal = TESOL Journal. 7(4):898–920, 2016.

[MIm] Crossword Generationn Tool. https://www.crosswordconstruction.com/ [last access: 2025-08-28] Abstract presented at NCUR 2021.

[MLG90] M. Halpin P. Torrance C. Mark M. L. Ginsberg, M. Frank. Search Lessons Learned from Crossword Puzzles. In *AAAI*, volume 90, pages 210–215, 1990.

[Msh20] V. V. Mshayisa. Students' perceptions of Plickers and crossword puzzles in undergraduate studies. *Journal of Food Science Education*, 19(2):49–58, 2020.

[MT13] T. Bastiaens S. Stijnen M. Thurlings, M. Vermeulen. Understanding feedback: A learning theory perspective. *Educational Research Review*, 9:1–15, 2013.

[Nic11] R. S. Nickerson. Five down, Absquatulated: Crossword puzzle clues to how the mind works. *Psychonomic bulletin & review*, 18(2):217–241, 2011.

[NK22] H. Bulut A. Ay N. Kalkan, S. Güler. Views of students on the use of crossword and word search puzzle as a teaching technique in nursing education: A mixed-method study. *Nurse Education Today*, 119:105542, 2022.

[RH21] S. Heinicke R. Heinen. Gestaltung von Lernmaterial und Didaktische Typografie–wie sich die Lesbarkeit von Texten auch ohne sprachliche Anpassungen verändern lässt. *PhyDid B-Didaktik der Physik-Beiträge zur DPG-Frühjahrstagung*, 2021.

[RPS11] Henry L. Roediger III, Adam L. Putnam, and Megan A. Smith. Chapter One - Ten Benefits of Testing and Their Applications to Educational Practice. volume 55 of *Psychology of Learning and Motivation*, pages 1–36. Academic Press, 2011.

[SN21] U. Schroeder S. Noichl, S. Korth. Inklusives und handlungsorientiertes lernen mithilfe digitaler gamebooks. In *DELFI 2021*, pages 145–150. Gesellschaft für Informatik e.V., Bonn, 2021.

[SN23] U. Schroeder S. Noichl. Interaktionsmöglichkeiten und Potenziale digitaler Gamebooks unter Berücksichtigung der CELG-Taxonomie, 2023.

[SP18] V. P. Giri D. Datta P. Kumawat P. Singh P. S. Matreja S. Patrick, K. Vishwakarma. The usefulness of crossword puzzle as a self-learning tool in pharmacology. *Journal of Advances in Medical Education & Professionalism*, 6(4):181, 2018.

[Sri14] S. Sridevi. User Interface Design. *International Journal of Computer Science and Information Technology Research*, 2(2):415–426, 2014.

[ST21] E. Wascher D. Schneider S. Thönes, S. Arnau. Boosting working memory with accelerated clocks. *NeuroImage*, 226:117601, 2021.

[STN23] J. A. Purohit S. T. Naik. Effectiveness of e-Crossword Puzzle tool in the Multidisciplinary course for the undergraduate students. In *Proceedings of the 16th Annual ACM India Compute Conference*, COMPUTE '23, page 37–42, New York, NY, USA, 2023. Association for Computing Machinery.

[visa] Alverde Magazin dm. https://www.dm.de/unternehmen/alverde-magazin/gewinnspiele/kreuzwortraetsel-175330 [last access: 2025-09-07].

[visb] raetsel.ch. https://raetsel.ch/online-demos-kreuzwortraetsel/ [last access: 2025-09-07].

[w3s] JS RegExp. https://www.w3schools.com/js/js_regexp.asp [last access: 2025-08-15].

[Wik] JavaScript/Objekte/RegExp. https://wiki.selfhtml.org/wiki/JavaScript/Objekte/RegExp [last access: 2025-08-15].

[Zho19] Y. Zhonggen. A Meta-Analysis of Use of Serious Games in Education over a Decade. *Int. J. Comput. Games Technol.,* 2019:4797032:1–4797032:8, 2019.

Eidesstattliche Versicherung
Declaration of Academic Integrity

Müller, Silvia Alexandra
Name, Vorname/Last Name, First Name

445514
Matrikelnummer (freiwillige Angabe)
Student ID Number (optional)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende ~~Arbeit~~/Bachelorarbeit/ ~~Masterarbeit~~* mit dem Titel
I hereby declare under penalty of perjury that I have completed the present paper/bachelor's thesis/master's thesis* entitled

Crossword Puzzles in Digital Education with Gamebooks

selbstständig und ohne unzulässige fremde Hilfe (insbes. akademisches Ghostwriting) erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt; dies umfasst insbesondere auch Software und Dienste zur Sprach-, Text- und Medienproduktion. Ich erkläre, dass für den Fall, dass die Arbeit in unterschiedlichen Formen eingereicht wird (z.B. elektronisch, gedruckt, geplottet, auf einem Datenträger) alle eingereichten Versionen vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
independently and without unauthorized assistance from third parties (in particular academic ghostwriting. I have not used any other sources or aids than those indicated; this includes in particular software and services for language, text, and media production. In the event that the work is submitted in different formats (e.g. electronically, printed, plotted, on a data carrier), I declare that all the submitted versions are fully identical. I have not previously submitted this work, either in the same or a similar form to an examination body.

Aachen, September 8, 2025
Ort, Datum/City, Date

S. Müller
Unterschrift/Signature

*Nichtzutreffendes bitte streichen/Please delete as appropriate

Belehrung:
Official Notification:

§ 156 StGB: Falsche Versicherung an Eides Statt
Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
§ 156 StGB (German Criminal Code): False Unsworn Declarations
Whosoever before a public authority competent to administer unsworn declarations (including Declarations of Academic Integrity) falsely submits such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment for a term not exceeding three years or to a fine.
§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt
(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.
(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.
§ 161 StGB (German Criminal Code): False Unsworn Declarations Due to Negligence
(1) If an individual commits one of the offenses listed in §§ 154 to 156 due to negligence, they are liable to imprisonment for a term not exceeding one year or to a fine.
(2) The offender shall be exempt from liability if they correct their false testimony in time. The provisions of § 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:
I have read and understood the above official notification:

Aachen, September 8, 2025
Ort, Datum/City, Date

S. Müller
Unterschrift/Signature