Erschienen in: IEEE Transactions on Visualization and Computer Graphics. IEEE. 2025, 31(7), S. 4034-4049
https://doi.org/10.1109/tvcg.2024.3393012

TreEducation: A Visual Education Platform for Teaching Treemap Layout Algorithms

Johannes Fuchs, Bastian Jäckl, Michael Jüttler, Daniel A. Keim, and Rita Sevastjanova

***Abstract*—Treemaps are a powerful tool for representing hierarchical data in a space-efficient manner and are used in various domains, including network security or software development. However, interpreting the topology encoded by nested rectangles can be challenging, particularly compared to tree-structured representations like node-link diagrams or icicle plots. To address this challenge, we introduce *TreEducation* , a visual education platform designed to improve the visualization literacy skills required for reading treemaps among non-expert users. *TreEducation* is an online application that combines visualizations, interactions, and gamification elements to facilitate understanding of eight different treemap layout algorithms and enhance students' learning process. We evaluated *TreEducation* in a classroom setting and a controlled environment. Our results indicate a significant knowledge gain of students training exclusively with *TreEducation* and the usefulness of *competition* as a social gamification element included in our competitive quiz.**

***Index Terms*—Treemap, education, gamification, verbalization, teaching.**

I. Introduction

VISUALIZATION literacy is understanding, interpreting, and creating visual representations of data and information [1]. It is an essential skill that enables people to extract insights and patterns, making better decisions based on the information. Unfortunately, many struggle to interpret more complex visualization techniques like node-link diagrams or treemaps [2]. This is especially true for novices like undergraduate students [3]. Teaching how to read such visualization techniques is key to closing this knowledge gap.

However, the traditional classroom setting falls short when teaching treemaps. Lecturers have to prepare abundant material, mainly slides, to illustrate the different steps for creating the visualization and what effect these steps have when applied to some data examples. Typically, they complement the prepared material with drawings on a flipboard. Such a setting lacks interactivity, as the example cases have to be pre-arranged. There is no straightforward way to explore the consequences of alternative design decisions and the final visual result on different kinds of data. As a result, students have to cope with a high level of abstraction.

To tackle this issue, interactive visualization systems have been introduced to support learners and lecturers in their endeavors. Prominent examples are algorithm visualizations to explain basic statistics [4], sorting algorithms [5], or tree traversal strategies [6]. In such settings, interaction and a step-wise procedure like animation chunking were deemed useful when explaining algorithms [7]. Also, the use of visualizations was attributed to having a positive effect on students' learning success [8]. Such education systems are highly tailored toward their specific domain and cannot be easily extended. That is why learning to read or understand some specific visualization techniques or algorithms is not yet supported by existing education tools.

Treemaps, for example, are a type of data visualization that display hierarchical relationships with nested rectangles [9]. These rectangles represent nodes in the hierarchical structure. Their position depends on the chosen layout algorithm, and they can change in size or color to convey additional information. Currently, only one tool has been published trying to improve people's treemap visualization literacy skills [10]. The software links a node-link diagram with a treemap constructed with the Slice-and-Dice algorithm. Interactive manipulation allows linking and brushing by hovering over nodes or rectangles, or details on demand by clicking on rectangles. Via a menu, seven precompiled datasets can be explored in the software, and six different color ramps can be applied to the treemap to visualize the underlying data value. The authors conducted a quantitative experiment that proved the benefits of the software compared to traditional teaching material. Unfortunately, the software is currently difficult to share across institutions since it is unavailable online. Moreover, the users' motivation to use such an educational tool in practice needs to be clarified.

In this paper, we introduce our online education software *TreEducation*,[1] a gamified learning platform that uses visualization, interaction, and verbalization elements to overcome the outlined didactic issues when teaching treemaps and gamification to strengthen the users' motivation to use the tool for learning purposes. We took Firat et al.'s [10] work as a starting point and tried to improve the software by introducing multiple layout

Manuscript received 10 September 2023; revised 9 April 2024; accepted 16 April 2024. Date of publication 24 April 2024; date of current version 20 June 2025. This work was supported in part by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2117 under Grant 422037984, and in part by the Federal Ministry of Education and Research (BMBF) in VIKING under Grant 13N16242. Recommended for acceptance by R. Metoyer. *(Corresponding author: Johannes Fuchs.)*

Johannes Fuchs, Bastian Jäckl, Daniel A. Keim, and Rita Sevastjanova are with the University of Konstanz, 78464 Konstanz, Germany (e-mail: johannes.fuchs@uni-konstanz.de; bastian.jaeckl@uni-konstanz.de; keim@uni-konstanz.de; rita.sevastjanova@uni-konstanz.de).

Michael Jüttler is with the Zürich University of Applied Sciences, 8400 Winterthur, Switzerland (e-mail: juet@zhaw.ch).

[1] https://treeducation.dbvis.de

algorithms, further design variations, alternative explanation methods, more didactic support, and directly integrated quizzes.

TreEducation was evaluated in an actual classroom setting and in a controlled environment. Based on already established treemap literacy tests, our results indicate a significant knowledge gain for students when training with *TreEducation* . This is true for different layout algorithms like Slice-and-Dice, or Strip. Also, the included competitive quiz proved the usefulness of social gamification elements, like *competition*.

II. Background

Games are typically characterized by rules, competition, and a clear outcome or goal that human participants achieve. Gamification refers to the integration of game design elements into non-game contexts [11]. This distinguishes game-based applications from serious games, which incorporate multiple game elements to create a complete game. The application area for gamification elements is not limited to entertainment. They can be applied in various contexts, such as health [12], marketing [13], visual analytics [14], and education [15], to improve user experience and engagement.

A. A Taxonomy of Gamification Elements

Gamification has been researched extensively in the past. As a result, different ways how to categorize single gamification elements exist [16], [17], [18]. Our work focuses on the taxonomy proposed by Toda et al. [19] since they specifically focused on the education domain. Additionally, the authors derived their taxonomy and 21 gamification elements after consulting experts from different professions like game developers, teachers, or researchers [20]. Based on their classification, these 21 gamification elements fit into five different categories.

Performance is a category that provides feedback to the user through different means. *Points* or *acknowledgements* can be used as extrinsic feedback to show achievements. *Statistics* or *progression* provide guidance about current advances and act as orientation in the virtual environment. Students might get confronted with more complex tasks depending on their *level*.

Ecological relates to the education platform interacting with the user. Students experience different content based on their *imposed choices* or gain *rare* items, which can be traded in the system's *economy*. Results based on user interaction can happen by *chance* and can be restricted by *time pressure*.

Social is the context that describes interactions of learners in the environment. Students can *cooperate* to achieve a common goal or *compete* against each other. In either case, gaining *reputation* is a way to represent social status and establish a hierarchy. This can result in *social pressure* for the learners.

Personal relates to the learner using the education platform. Motivation can be increased with *novel* content, new information, or updating game elements triggering different levels of *sensation* like the visual system. Overall, students must have a clear *objective* or challenges like *puzzles* they can fulfill. If they fail a task, *renovation* is necessary by repeating tasks or transferring knowledge from already known contexts.

Fictional ties the user experience with the environment. *Storytelling* supports the narrative of the education platform through visual, audio, or sensual cues. The order of events happening in the story is called the *narrative*. Users can influence this order through implicit choices.

In Section III, we will evaluate educational tools based on this taxonomy.

B. Benefits and Drawbacks of Gamification Elements

Gamification elements often constitute an inseparable component within digital teaching and e-learning [21]. The reasons for this are that they originated from digital video games (e.g., Sailer et al. [22]) and are attributed with great potential for enhancing engagement, motivation, and attention during task performance [19], [23]. As a result, numerous research groups have studied the influence of gamification elements in digital learning for many years [24]. A summary can be found in the systematic literature reviews [25], [26], [27]. The essential elements identified include points, digital badges, and leaderboards. These three gamification elements are often called the *PBL triad* [22], positively impacting learning outcomes. Additionally, they can be easily implemented into existing e-learning platforms [27].

Based on the meta-analysis by Sailer and Homner that includes 38 publications with experimental studies between 2013 and 2017, gamification elements show a significant positive effect on cognitive learning outcomes that comprises students' conceptual and application-oriented knowledge [21]. Regarding motivational outcomes (e.g., intrinsic motivation, engagement, or attitude) and behavioral skills (e.g., technical skills or motor skills), results of the meta-analysis showed that gamification elements are most effective in a competitive-collaborative setting, which means that students could, e.g., compete in learning groups. This comprehensive meta-analysis underpins the great potential of gamification.

C. Gamified Learning versus Serious Games

Regarding the use of games in teaching-learning arrangements, different terms and concepts exist in the literature [11], [28]. First, a distinction is made between the terms "play" and "game", where the term "play" describes any form of playful interaction, while playing "games" refers to a concrete area. "Games" thereby combine different characteristics. According to Loh et al. [29] and Mayer [30], games are rule-based simulative systems that are responsive, cumulative, challenging, and inviting. Accordingly, games integrate pre-defined rules in which players make decisions. These decisions lead to reactions (responsive) that change their status within the game (cumulative). Furthermore, decisions should not be predetermined by the game (challenging) but inviting to the player (e.g., by gaining rewards). As described in the taxonomy of gamification elements, games consist of different elements (Section II-A).

Theoretically, these elements can be distinguished into elements that purely serve the purpose of entertainment (e.g., leveling up an avatar; entertainment games) or serve the purpose of a certain learning goal (e.g., making decisions in order to solve an authentic problem). Gamification (or gamified learning) makes use of one or some of these elements by implementing them in a non-game context (e.g., a teaching-learning arrangement at a university) [11]. Gamified learning does not provide a whole game that students play but uses single gamification elements to benefit from them. In contrast, serious games (or game-based learning) provide a complete game that uses these elements to fulfill entertainment and/or learning goals.

In summary, *TreEducation* is a gamified learning platform using different gamification elements to impact learning and motivational outcomes positively. The presented taxonomy will allow for an objective evaluation of *TreEducation* against already established education tools presented in Section III.

III. Related Work

The implementation of *TreEducation* was inspired by research from two different domains. First, we reviewed research about treemap visualization techniques and their use in practice. We wanted to identify well-established designs together with promising variations to reproduce the most crucial treemap representations. Second, we focused on educational tools to enhance visualization literacy. Since *TreEducation* is meant to support students and teachers in their endeavors, we wanted to understand well-established education methods in game-based learning and evaluate whether they can be transferred to our specific domain.

A. Treemap Visualization Techniques

Treemaps have received a lot of research attention. Many applications have been introduced, which stem from entirely different domains like network security [31], biology [32], geography [33], or sports [34]. Recent surveys reviewed the effectiveness of treemaps [35], categorized possible layout strategies [36], or came up with an overall taxonomy [37].

According to Scheibel et al. there is a distinction between space-filling treemaps, which exploit the entire screen space, and containment treemaps, which introduce empty space to communicate the hierarchical structure better [37]. A famous example of the latter is PhotoMesa [38]. Currently, *TreEducation* focuses on space-filling treemap algorithms. However, the application can be easily extended to include further variations since the learning components and interactive features are independent of the treemap algorithm.

Furthermore, treemaps differ in how the rectangles are positioned. Scheibel et al. systematically reviewed and categorized different layout algorithms [36]. They concluded that rectangular two-dimensional layouts are used most often. Since positioning the rectangles is an optimization problem, alternative approaches exist for optimizing for different criteria. To reach a broad audience, *TreEducation* contains the well-established rectangular splitting layout Slice-and-Dice [9]. However, since this technique has been criticized, our software also includes the Squarify [39] and Strip arrangements [40]. These two algorithms improve the aspect ratio of the rectangles but are more difficult to read. An online application using these algorithms is newsmap, a tool to browse Google News [41].

The BinaryTree layout from the University of Maryland [42] and the Number-Balanced arrangement from Feng et al. [43] have been included for a more balanced partitioning of the nodes. They both allow for a good compromise between a solid aspect ratio and a good temporal coherence. Furthermore, the Moore curve [44] layout optimizes the continuity of the rectangles' positions, whereas the Greedy Insertion [45] aims for more regular layouts.

Layouts focusing more on time series than on hierarchical data like Fractal Figures [46], ID-Map [47], or Multiresolution Grid Layouts [48] are not considered. More exotic arrangements like three-dimensional, list-focused, or polygonal layouts can be implemented in the future since they also fit into our overall application design.

Different coloring strategies can be applied to the rectangles independently of the layout strategy. Color hue can be used to distinguish different categories in the data like geographic location [49]. Also, luminance can be varied to encode attribute values since it does not conflict with the size of the rectangles [50]. To communicate the hierarchical structure better, a color gradient can be applied [51]. It seems as if the use of color is highly task or application-dependent. Therefore, *TreEducation* allows the user to introduce color as he/she sees fit.

In summary, users can apply different rectangular splitting layouts on hierarchical data. For a simple evaluation, algorithms optimizing different criteria have been implemented in *TreEducation* . Color encoding can be added to the visualization to improve the hierarchy reading or encode additional information independent of the underlying layout algorithm.

B. Educational Tools and Visualization Literacy

Although *TreEducation* focuses on treemap literacy with specific learning objectives, our goal was to identify promising gamification elements (highlighted in italics) to transfer to our education platform. Therefore, this section does not focus on individual visualization techniques but on a holistic view of data visualization literacy. A detailed investigation of how well students understand novel data representations can be found in the state-of-the-art report from Firat et al. [52].

C'est la Vis is an application to teach basic visualizations like bar charts at elementary schools [53]. Students can choose between different scenarios (*narrative*) and experiment with different scales and abstraction levels (*imposed choice*). A progression bar and some general statistics act as orientation support (*progression, statistics*). Furthermore, students can test their skills in exercises created by teachers (*puzzle*).

Construct-a-vis is a tablet visualization tool implemented to support elementary school students in creating free-form visualizations [54]. Teachers can configure the application to fit the needs and abilities of the students (*narrative*). Students can create, modify, or combine tokens to represent a given data table (*imposed choice*). During creation, the data table provides visual feedback about the suitability of the chosen design (*sensation*).

TABLE I
ALLOCATION OF GAMIFICATION ELEMENTS: CELLS ARE COLORED IF AN APPLICATION USES GAMIFICATION ELEMENTS FROM THE RESPECTIVE CATEGORY

Tool	Performance	Ecological	Social	Personal	Fictional
C'est la Vis	progression, statistics	imposed choice		puzzle	narrative
Construct -a-Vis		imposed choice	cooperate	sensation	narrative
EduFeed	acknowledgement	imposed choice	reputation, competition	puzzle	
Visual Morphing		imposed choice		renovation	
Treemap Literacy		imposed choice		renovation, puzzle	narrative
Cheat Sheets				renovation	story-telling

Colors are chosen based on the taxonomy of gamification elements. The number of gamification elements is not encoded in this table. Social and performance based gamification elements are barely used.

Students may collaborate in a shared working space to solve the task *cooperatively*.

A more competitive approach is proposed in EduFeed [55]. After deciding on different activities (*imposed choice*), students work on exercises (*puzzle*) while competing against each other (*competition*). Their progress is recorded via badges (*acknowledgment*) and visible to others (*social pressure*, *reputation*).

More complex visualization techniques are taught through visualization morphing [56]. The idea is to teach an unknown visualization technique by morphing a previously known visualization (*renovation*). For example, a simple line chart is morphed into a spiral layout to teach circular time series visualizations. The user can interactively choose the sequence of morphing steps (*imposed choice*).

A similar idea is proposed by Firat et al. to improve treemap literacy [10]. A tree and a treemap are shown simultaneously on the screen. When learning about treemaps, students can relate to the easier-to-understand node-link diagram (*renovation*). The views can be manipulated interactively (*imposed choice*, *narrative*). After exploring the software, students may solve a predefined questionnaire to test their knowledge (*puzzle*). The application was further improved with additional interaction techniques like expanding nodes, a coloring scheme to communicate the extent of nodes, and a more visible connection between the node-link diagram and the treemap view [57].

Less interactive approaches introduce videos [58] or cheat sheets [59] to teach data visualization. Both approaches focus on telling a story to explain the underlying concept (*storytelling*). Whereas the videos show the same technique with different data, cheat sheets use familiar representations to teach unknown visualizations (*renovation*).

Table I provides a summary of gamification elements used in the different education platforms to improve visualization literacy. Although research has shown the usefulness of performance measures like points or social gamification elements like competition, they are barely implemented in the presented visualization education tools. *TreEducation* offers quizzes to collect points to see the progression and allows for competition among learners.

IV. EDUCATIONAL OBJECTIVES

Studies using game-based or gamified learning often neglect learning theories [60]. However, theoretical foundations are important to explain the effects of games or gamification elements on students' learning processes and outcomes. Based on their meta-analysis, Wu et al. find that most studies in game-based learning explain learning as an individual process of experience in which students construct their cognitive representation of a learning object and in which learners interact with their social environment [60]. These considerations strongly follow those represented in modern learning theories of humanism and constructivism [61]. Other learning theories, such as cognitivism or behaviorism, are mostly neglected.

Against this background, Wu et al. strongly recommend combining ideas of both constructivism and cognitivism to explain student learning within game-based or gamified learning environments [60]. In our study, we are following two theoretical approaches to formulate educational objectives (**EO**). First, following the elaboration theory of instruction by Reigeluth, student learning should follow a simple-to-complex sequence to ensure that the learner is always aware of the context and importance of the different ideas being taught [62]. Accordingly, the learning content must be structured from simple to complex situations or problems.

Second, following constructivist approaches, such as situated learning theory, student learning can be described by simple cognitive re-organizations and representations [63], [64]. Learning is always an individual process where students are actively engaged and socially interact with their learning environment. This consideration shifts the focus from simple instructive to constructive learning. Learning is thus characterized by:

- **EO1:** A high degree of learner autonomy [65].
- **EO2:** The solution of authentic problems [66], [67].
- **EO3:** Learning in multiple contexts [68].
- **EO4:** Social interaction, in which learners are not passive recipients but active participants [67].
- **EO5:** Systematically embedding new knowledge structures with increasing complexity into their existing cognitive structures [62].

Our learning approach with *TreEducation* fulfills those educational objectives. Students can decide for themselves how complex the problem to be solved should be (**EO1**). Students learn by solving authentic problems (**EO2**) at their own pace (**EO1**). Teachers also have the opportunity to teach not only abstractly, through pure instruction, but to present the learner with variable problems (**EO2**) and let them solve them cooperatively or on their own (**EO4**). This allows students to solve problems differently and discover solutions (**EO3**). With this gamified and explorative approach, students can connect new knowledge structures to existing ones at different levels (**EO5**).

V. THE *TREEDUCATION* APPLICATION

TreEducation was built around those educational objectives. The software was implemented using the JavaScript library D3 [69] and is connected to a PostgreSQL database. It has six major components (Fig. 1). The user can interact with these

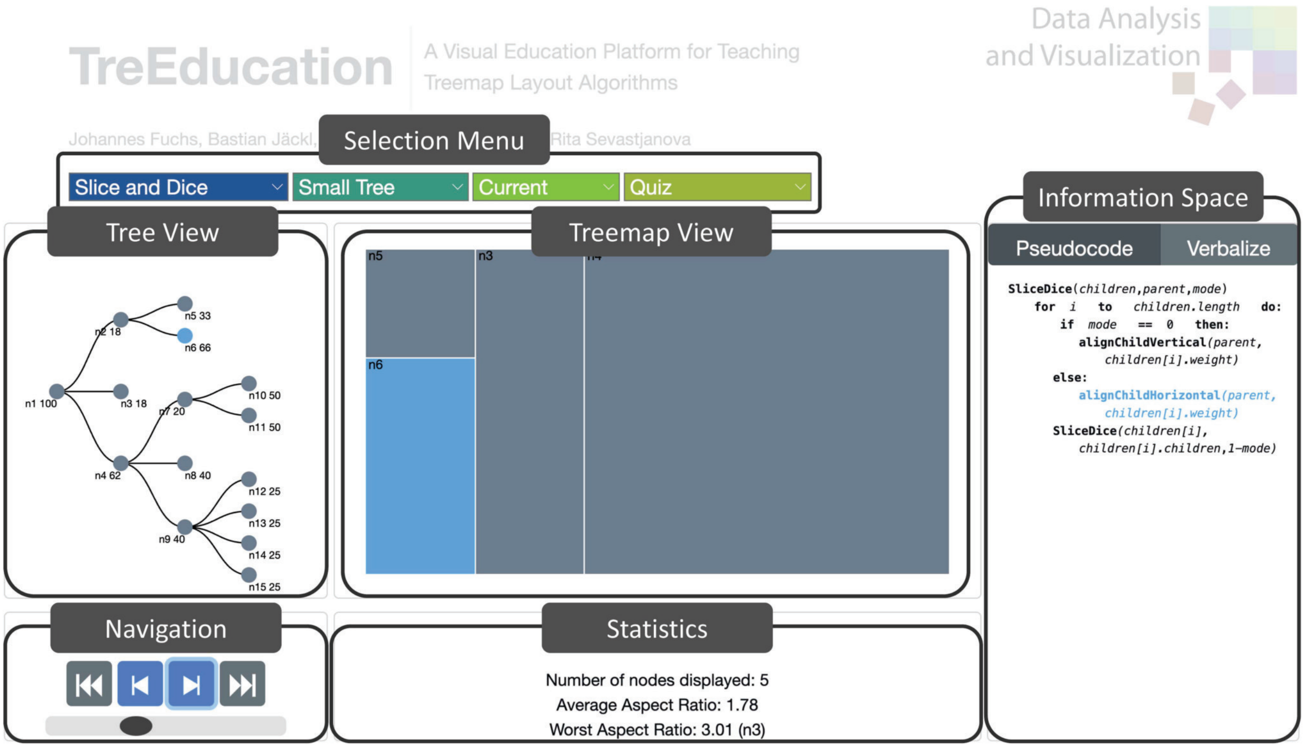

Fig. 1. *TreEducation's* six major components are artificially highlighted. Node n6 has been positioned in the *Treemap View* using the Slice-and-Dice layout algorithm chosen from the *Selection Menu*. The current node is highlighted in the treemap and the node-link diagram shown in the *Tree View*. The *Information Space* on the right contains the pseudocode, which emphasizes the current step taken in the algorithm. The *Statistics* at the bottom provides useful information about the current status of the treemap. Learners can interactively continue to build the treemap using the *Navigation* in the lower left corner. For each component, tooltips are available to provide additional information.

components freely to allow for a self-determined learning experience (**EO1**). The application scales well to a high number of accesses since all calculations are performed on the client side.

A. *Selection Menu*

The menu (Fig. 1, top) comprises four drop-down lists that enable the selection of algorithms, datasets, color encodings, or the option to access interactive quizzes.

Algorithms: We discuss eight different layout algorithms belonging to the group of rectangular splitting layouts [36]. The algorithms have been chosen to cover diverse aesthetic criteria like aspect ratio, order preservation, or balance. Although this collection of algorithms serves as a solid foundation for introducing layout techniques, it is important to note that it is not an exhaustive list. We intend to expand this selection in the future to encompass alternative methods.

Datasets: The initial pool of datasets was derived from literature [9], [39] as well as from our experience in teaching and data analysis. These datasets encompass a range of diverse characteristics, including variations in the number of data points, their weights, the depth of the hierarchy, and the overall distribution of relationships. Providing good examples of datasets to show the algorithmic behavior effectively supports learners in their endeavors (**EO2**) [70].

In cases where users find the current dataset selection insufficient, they can manually create a node-link tree by adding data points with different weights and relations onto an empty 2D plane. The newly created dataset can be saved and utilized with any layout algorithm. Including this custom data functionality empowers lecturers to promptly respond to student inquiries in the classroom (**EO4**) and demonstrate the distinctive behaviors of various algorithms using customized data. Also, students can create infinite datasets to evaluate layout algorithms working with different data characteristics (**EO1, EO5**).

Color encodings: Rectangles can be colored to represent additional information. *TreEducation* offers five different color settings. By default, the latest rectangle added during the construction of the treemap is highlighted. This coloring feature supports users during the step-by-step positioning, focusing on newly created rectangles.

To improve the understanding of the hierarchical structure, rectangles can be color-coded based on their subbranch. A maximum of ten categorical colors are utilized to ensure clear visual differentiation, leveraging D3's schemeCategory10 functionality. Another way to facilitate the hierarchy reading is by introducing cushions [51]. Each rectangle is encoded with a color gradient to increase the contrast with neighboring rectangles. Brighter colors correspond to deeper hierarchy levels.

The fourth option is to map color to different data characteristics, like the nodes' order or the rectangle's aspect ratio. This allows users to evaluate layout algorithms quickly. For example, a consistent gradient across the entire treemap indicates

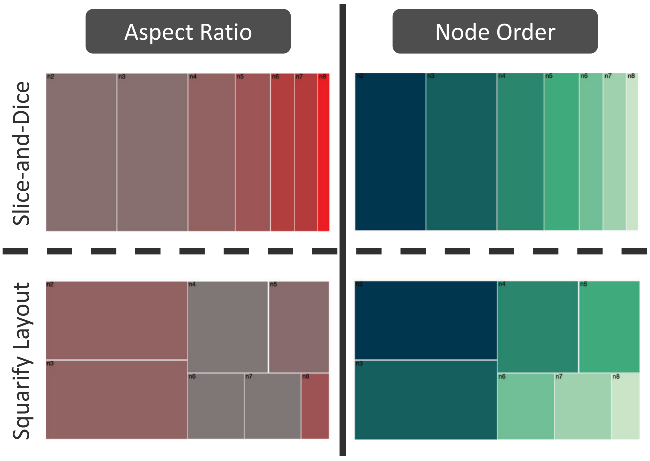

Fig. 2. *Color options:* The same dataset is being displayed with the Slice-and-Dice algorithm (top row) and the Squarify layout (bottom row). The two algorithms can be easily compared using different color schemes like aspect ratio (left) and order preservation (right).

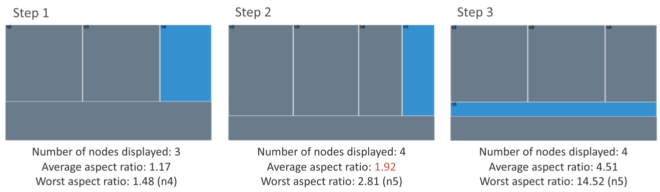

Fig. 3. *Strip Layout:* Screenshot from the Treemap View and the corresponding Statistics. In step 2, node n5 will be positioned at the current row. However, since the average aspect ratio worsens (artificially highlighted) compared to the previous step (step 1), the rectangle will be removed, and a new row will be created where the rectangle will be placed (step 3).

well-preserved node order (Fig. 2). If rectangles are colored red, the aspect ratio is unbalanced (Fig. 2).

Interactive quizzes: Students can test their knowledge with three different kinds of quizzes directly integrated into the software (**EO3, EO4**). The selection menu allows users to choose between a jigsaw puzzle, a randomized quiz, and a competitive quiz. The supported approaches are described in Section V-F.

B. Tree View

A node-link diagram with a tree layout is displayed to show the hierarchical data. This representation conveys the topological structure better in comparison to treemaps [71]. Therefore, students do not need additional training to understand the underlying data. The nodes in the tree are linked to their corresponding rectangles in the treemap. That is why a previously selected color encoding is applied to the nodes, too. Also, users can hover over nodes for a brushing & linking effect. This connection helps to recognize entities in the two visualizations (*renovation*). To further support the identification of nodes, labels are added to indicate the weight and identifier as suggested by Firat et al. [10].

C. Treemap View

This component is the major view of the *TreEducation* application. It displays the treemap, which depends on the layout algorithm, the data, and the visual encoding chosen in the selection menu (Section V-A). The treemap has a resolution of $450x250$ pixels in size. At first, the treemap consists only of the root node. Rectangles are added to the view depending on users' interactions with the navigation area (Section V-E) (**EO4**). This incremental construction via a step-by-step process is proven to be effective for understanding algorithmic behavior like different layout algorithms (*progression*) (**EO5**) [70]. Also, since the view is linked to the node-link diagram and the current node is highlighted in both displays, students experience a transformation from the node-link view to the treemap similar to visualization morphing (*renovation*) [56].

After adding the first rectangle to the root, summary statistics are displayed underneath the treemap. These comprise the current number of nodes visible in the treemap, the average aspect ratio, and the worst aspect ratio with the respective node's identifier. This additional information helps to evaluate different layout algorithms or to understand certain decisions an algorithm makes when positioning a rectangle.

For example, the Strip layout [40] tries to improve the aspect ratio based on the current average aspect ratio. Therefore, it has to decide whether a rectangle should be positioned in the current row or if a new row has to be created Fig. 3. Thus, positioning a single node in the treemap involves multiple steps experiential in *TreEducation* . First, the algorithm will position the rectangle in the current row and calculate the new average aspect ratio. Then, the aspect ratio is compared to the previous average aspect ratio. If the aspect ratio improves, the rectangle will be kept. If the aspect ratio worsens, the rectangle will be removed from the current row, and a new row will be created where the rectangle will be inserted. The user can experience this entire sequence of steps accompanied by the relevant statistics and the dynamic pseudocode or verbal explanation (Section V-D).

D. Information Space

Inspired by previous work on education software solutions like EduClust [6], [72], we incorporated the pseudocode of different layout algorithms to explain the algorithmic behavior. The information space presents the current layout algorithm in pseudocode (**EO3**). The algorithmic steps are shown in a structured way, being human-readable even without programming knowledge. The static pseudocode is enriched with dynamic content to improve the understanding further. Learners can hover over individual components of the pseudocode to get additional information. This option allows students to trace the algorithmic flow in control structures. E.g., variables are resolved to their actual values to simplify the understanding of decisions made in certain conditions (Fig. 4).

If this description is still too abstract, the information space offers the possibility to switch to a verbalization of the individual steps of the layout algorithm (*imposed choice*) (**EO3**). Verbalization or natural language explanations are often used to explain algorithmic behavior (see, e.g., [73], [74], [75]).

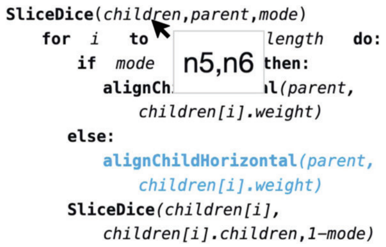

Fig. 4. *Dynamic Pseudocode:* A tooltip provides additional information about variables and functions mentioned in the pseudocode. In this example, the user hovers over the variable *children* to see the nodes in this container (i.e., n5 and n6).

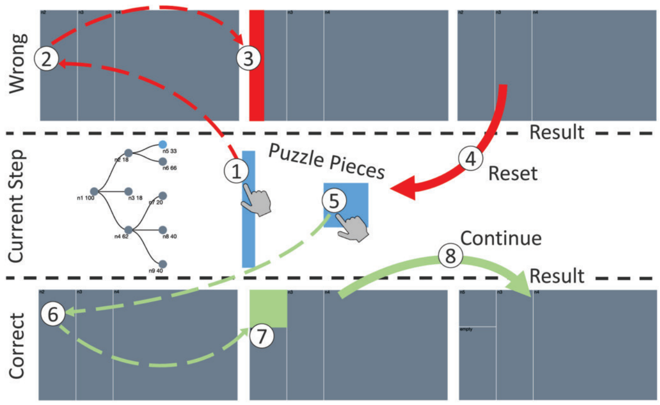

Fig. 5. *Jigsaw Puzzle:* Users must drag and drop puzzle pieces to their correct location. The current node which has to be positioned is highlighted in the node-link diagram. There is always the choice between two alternatives (step 1 and 5). If the location is chosen wrongly, the system automatically provides visual feedback (step 2 and 3) and resets the components (step 4). If the location is correct, the system responds with green colors, locks the puzzle piece (step 6 and 7), and continues with the next question (step 8).

It has been shown that in comparison to other explanation methods (e.g., visualizations), they are more plausible (i.e., convincing) [76] and they increase the accessibility for different target user groups [77]. According to Ehsan et al. [73], explanations in natural language offer several advantages compared to other explainability methods. In general, explanations per se are grounded in natural language communication. Thus, they are more easily understood by humans, resulting in increased satisfaction, confidence, and a greater willingness to utilize autonomous systems (*storytelling*).

E. Navigation Area

To steer the treemap construction, users can navigate the automatic layout process with familiar icons and options (*imposed choice*) (**EO1**). Common media players inspire our design and interaction concept. Users can step through the process of layouting rectangles. In each step, a single rectangle is been positioned. Additionally, users can return to previous states and experience former steps again (*renovation*). An interactive slider can be used to browse through the entire history of algorithmic steps quickly.

We decided not to animate the individual steps so learners could experience the algorithm at their own speed. However, creating the entire treemap at once is possible if they are more interested in the final result. This feature allows learners to evaluate entire treemaps efficiently.

F. Jigsaw Puzzle and Quizzes

The treemap jigsaw puzzle allows users to construct treemaps via a drag-and-drop interface incrementally (*puzzle*). In each step, the jigsaw puzzle offers two different-sized rectangles (Fig. 5). Users must choose one and drag it to the final destination (*imposed choice*). The correct answer depends on the size of the rectangle and the underlying layout algorithm. A snapping feature supports the user during the construction, allowing for minor positioning errors. After dropping the rectangle, the application responds to correct answers with a green color coding for the rectangle (*sensation*), providing positive conditioning [78]. According to the self-determination theory, this extrinsic motivation provides gratification and satisfies the need for competence. Wrong answers reset the current choices, and users can try again (*renovation*). Statistics about the number of correct placements and the overall number of attempts are shown to provide feedback to the student (*points*). Compared to video instructions or reading literature, research has shown that hands-on puzzles are more engaging and provide a better learning outcome [79].

The second quiz provides users with an infinite set of random questions. Those random elements offer a new experience each time the quiz is played (*novel, chance*). At the beginning, the user selects the algorithms he or she wants to answer questions about. After the selection, the quiz starts in the first of three levels. The starting questions are about reading and comparing sizes of rectangles embedded in treemaps constructed with one of the selected algorithms. The quiz then modifies its difficulty level in response to the player's performance (*narrative*). In the next level, further questions about the topology of the data and the relationship between data points will be asked. The last level increases the complexity of the data. Therefore, the quiz will dynamically vary to engage users and provide them with a continuous challenge (*level*) (**EO5**). The difficulty level is shown to the user as an orientation to rate their performance (*progression*) together with the ratio of correct and incorrect answers (*points*).

The last option is a competitive quiz with a predefined set of questions (*objective*). Questions are derived from literature but adapted to different datasets and layout algorithms and extended to topological tasks (**EO2**) [80], [81]. Answers are recorded and stored in a database. The ratio of correct answers per question is shown to the player (*points*). Students can compete with each other by comparing their scores with other students (*competition*). Social groups can be formed by typing in a four-digit identification number. Only the results of a pre-selected social group are shown to the user.

G. Gamification Elements

In summary, *TreEducation* incorporates 14 different gamification elements from all five categories introduced in Section II-A.

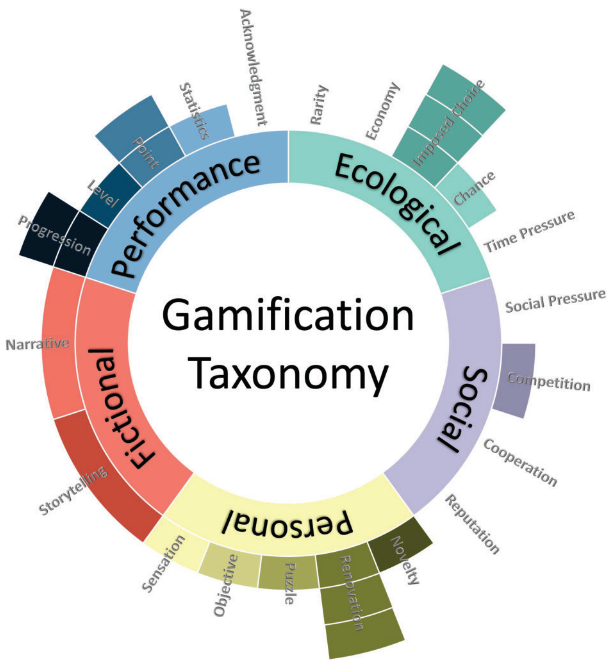

Fig. 6. *Gamification Taxonomy:* Overview of the different gamification elements and their amount of use in *TreEducation.* Color encodes the different categories, whereas the number of slices indicates the application of each gamification element individually.

As seen in Fig. 6, social aspects are the least supported elements. Since students can explore the application freely and choose how to proceed in their learning process, *imposed choice* is strongly represented. Also, familiar visualizations help to understand new representations, the stepwise layouting process allows students to go back and forth, and finally, quizzes can be repeated when giving wrong answers. The software, therefore, supports learners in transferring already known concepts and knowledge (*renovation*).

The possibility to solve puzzles and play quizzes offers the majority of gamification elements. Besides competition, students can also work on solving an *objective* when working on the official quiz. Since, in education domains, the lack of objectives and progression decreases students' motivation [82] incorporating different quizzes seems to be beneficial.

VI. Usage Scenario

Both lecturers and students can profit from *TreEducation* . Therefore, in our usage scenario, we showcase a visualization lecture about an introduction to treemaps and how the software can be used for preparation, execution, self-study, and review of the lecture content.

First, the lecturer decides on the specific layout algorithm she wants to teach. In our imagined scenario, she decides on the Slice-and-Dice algorithm as a simple start to the topic. To minimize preparation time, she uses *TreEducation* to set up her slides quickly. First, she selects the respective layout algorithm and an available dataset. To introduce the overall topic, she instantly calculates the resulting treemap in the navigation area (Section V-E). The visualization is then copied to the slides from the *Treemap View* (Section V-C) to explain the overall idea of treemaps. As a next step, she copies the pseudocode from the *Information Space* (Section V-D) to explain the general idea of the layout algorithm.

In class, she does not rely solely on slides or drawings on the blackboard but explains the algorithmic behavior with *TreEducation* . Like in the preparation, she selects the Slice-and-Dice algorithm and an available dataset. However, this time she executes the algorithmic sequence step by step using the navigation area. Therefore, the treemap is built incrementally. The current node is also highlighted in the *Tree View* (Section V-B) to support the understanding of the topological structure of the data. To underline her verbal explanations of each step, she hovers over the dynamic pseudocode to show additional details about the nodes and variables.

After running through the entire algorithm, she invites students to try *TreEducation* on their own by running the same algorithm with a different dataset. After familiarizing themselves with the software and the algorithm, students are encouraged to try an interactive quiz, specifically the jigsaw puzzle (Section V-A). In this setting, students can apply their knowledge directly and create the treemap with a drag-and-drop interaction, given the underlying layout algorithm. The software provides direct feedback about their performance.

At the end of the session, the lecturer summarizes the advantages and disadvantages of the layout algorithm, referring to the visualization in the Treemap View together with the corresponding *Statistics*. As homework for the following week, students are asked to try a different layout algorithm, i.e., Squarify, and compare it with the Slice-and-Dice algorithm. Since *TreEducation* is available online, students can repeat the steps from the lecture with the Squarify algorithm at home. To facilitate the comparison between multiple layout algorithms, *TreEducation* allows users to visualize multiple treemaps with their corresponding statistics on the same data using different algorithms. Having this side-by-side view facilitates the evaluation of different layouts and helps students in solving their homework.

In preparation for the exam, students use *TreEducation* to review treemaps. To check their knowledge, they decide to play an interactive random quiz, which automatically adapts the difficulty due to their performance. Since some students perform poorly, they revisit the step-wise construction of treemaps using their own datasets to have a bigger set of examples. After this additional preparation phase, they now want to compete against other students and decide to play the competitive quiz. Since they can see the performance of other students and also know that their answers get recorded, they can experience social pressure similar to an exam condition. Based on their performance in this quiz, students can decide to continue training using the software and be more confident about their knowledge.

VII. Evaluation

TreEducation contains different interactive features and gamification elements. To evaluate the suitability of the software for the learning process, we investigated parts of the toolset separately. We were most interested in our *verbalization* feature to simplify the explanation of algorithmic behavior and the consequences of introducing *competition* as a social gamification element in an education environment. Therefore, we came up with two separate user studies.

A. Verbalization versus Pseudocode

The user experiment was conducted in a classroom setting during tutorials accompanying an introductory lecture on data visualization. Among others, the course curriculum comprises the understanding of two different layout algorithms for treemaps, namely Slice-and-Dice [9] and Squarify [39]. Students are expected to be able to retrieve and compare values encoded in treemaps and describe the topological structure of the underlying data. Being able to implement the algorithms is not part of the curriculum.

1) Experiment Design: The experiment spanned over two weeks. In the first week, we applied a pre-test/post-test design to research differences in knowledge gain of students under two conditions: *verbalization (VB)* or *pseudocode (PC)*. Students were split into two groups working with either *VB* or *PC*, resulting in a between-group design. In the first session, students were expected not to have prior knowledge about treemaps and, specifically, the Slice-and-Dice algorithm. Participants in the *VB* condition got a verbal explanation of the underlying algorithm, whereas students in the *PC* condition had to rely on a dynamic pseudocode (Section V-D).

Between the first and the second sessions, participants had the chance to attend a data visualization lecture focusing on the Slice-and-Dice algorithm and the Squarify layout. Besides learning how to read the topological structure created by the two algorithms, the lecture also focused on comparing the two techniques. Therefore, the concept of a good aspect ratio was also introduced.

In the second session, participants were not split into two groups. All students followed the exact same procedure, answering questions concerning the Strip layout [40], which is not part of the curriculum. Also, another version of *TreEducation* was made available, which contained both the *pseudocode* of the algorithm and the corresponding *verbalization*. Students could switch between the two explanation methods via the information space menu tabs. We wanted to see which explanation method (*pseudocode* or *verbalization*) participants choose when being confronted with a new layout algorithm.

Procedure: At the end of the tutorial session, students could stay and participate voluntarily in the user experiment. As a prerequisite, students needed a laptop device with internet access to run the *TreEducation* software. The experimenter distributed the study material to each participant. The material comprised a consent form, an initial questionnaire, the pre-test, the post-test, and a final questionnaire. The experimenter was responsible for keeping track of the time and overseeing the experiment.

At the beginning of the study, participants signed the consent form and filled out the initial questionnaire asking for demographic information and their current knowledge level. After this introduction, the experimenter took the time. Participants had 10 minutes to work on the pre-test. They were not allowed to use any help to answer the questions about the Slice-and-Dice layout algorithm. After the pre-test, the experimenter granted participants access to the *TreEducation* software and showcased the application's basic features. Following this introduction, students got another 10 minutes to prepare for the post-test using a tailored version of *TreEducation* .

To only focus on the essential parts of *TreEducation* , features like selecting layout algorithms or switching between datasets were removed. As default, the Slice-and-Dice algorithm and a specific training dataset were selected. Also, options to play quizzes or customize the coloring of the treemap were unavailable. The only possible interaction was the stepwise creation of the treemap using the arrow buttons. The current node was highlighted in the tree and the treemap to allow for a visual link between the two representations. The information space either showed the algorithm's pseudocode or the verbal description. This choice depended on the between-group allocation of the participants (*PC* or *VB*). The post-test took as long as the pre-test (10 minutes) and contained the same questions; however, to avoid learning effects, the underlying data was exchanged. The experiment concluded with a final questionnaire asking 5-point Likert scale questions about the usefulness of individual features like the navigation area, or the color highlighting. Additionally, participants could provide feedback or comments as free text.

The procedure was identical for the second session. Except that the default algorithm was the Strip layout [40] and that both groups had access to the pseudocode and the verbalization component. They could switch between the two descriptions as they saw fit. The final questionnaire was adapted by swapping questions about individual features with questions about motivational predictors like perceived enjoyment or competence.

Tasks: We considered tasks provided by VLAT [80], miniVLAT [81], and the treemap literacy test from Firat et al. [10] to check the visualization literacy skills of participants. We had to reformulate certain tasks since we introduced space-filling, not containment treemaps. Also, questions about layout algorithms and the topological relationship between nodes were not sufficiently covered in the aforementioned tests. Therefore, our new task list ranged from simple size comparisons to reading topological relationships, understanding different layout algorithms, and constructing a treemap given a certain node-link diagram. The following list provides one example for each kind of task participants had to work on.

- *Size comparison* The number of unique visitors for <*node1*> was more than that of <*node2*> in 2010.
- *Topology* <*node1*> and <*node2*> are siblings.
- *Understanding* Which treemap (<*A* >, < *B*>, <*C*>) corresponds to the following node-link diagram?
- *Construction* Given the following node-link diagram, please sketch the corresponding treemap.

Data: We used four different data sources. First, we took the treemap shown in VLAT [80] to replicate parts of the literacy test and additionally asked questions about the topological relationship of nodes. Second, we used the node-link diagram introduced in the original paper from Shneiderman [9]. Participants had to pick the corresponding treemap out of a set of three. Third, we implemented a simple node-link diagram containing two inner nodes of equal weight on the first hierarchy level, each having two or three children, respectively. Users had to build a treemap given this tree with a specific layout algorithm. For the training phase with our *TreEducation* software, we took the node-link diagram introduced by Johnson and Shneiderman when explaining treemaps [83].

Participants: We recruited 23 participants (7 female, 12 male, and 4 NAs) from our data visualization lecture. Their age ranged from 20-35 years (median age 23). Twelve students reported a background in computer science, six were enrolled in the social economic data science track, and five did not report anything. 17 participants were familiar with hierarchical data, 12 with node-link diagrams, but only 4 with reading treemaps. In the first session, 20 students participated in the experiment. Eight of these 20 students also attended the second session, with three new students joining the participant pool. Therefore, we had a total of eleven students for the second session. Participants did not get any monetary compensation.

Hypotheses: Our hypotheses were derived from previous research on education software [72], verbalization [76], [77], and our experience as lecturers.

- **H1:** *Participants' performance regarding correct answers will improve between pre-and post-test when preparing with the* TreEducation *software. This is independent of the preparation condition (VB versus PC).* Studies have shown that the preparation phase supported with education tools between two tests will help participants to gain additional knowledge [72].
- **H2:** *Participants' knowledge gain will be higher in the VB condition compared to PC.* The knowledge gain is measured in terms of correct answers. Research has shown positive effects of verbalization in the training phase compared to more abstract explanation methods [84]. We expect this to be also true for understanding complex algorithmic behavior.
- **H3:** *Participants favor the VB condition compared to PC.* Verbal explanations are grounded in natural language and increase subjective preferences like satisfaction or confidence [73]. Therefore, learners are more likely to choose verbalization over more abstract explanations.
- **H4:** *Participants are motivated using* TreEducation *to learn about treemaps.* Previous research on education software for algorithmic behavior has shown a positive effect on participants' confidence and willingness to use such tools as preparation methods [72].

2) Results: We only report on significant results ($p < .05$) from our quantitative analysis and refer to the qualitative feedback in Section VII-A3. We used a Wilcoxon signed rank test with continuity correction to compare the difference in performance between the pre-and post-test. Participants' performance reflects the number of correct answers in each test.

First Session: Participants worked with the Slice-and-Dice algorithm and had to answer six different questions. Each question was worth one point, with six points resulting in 100 % correct answers (Fig. 7). The final questionnaire asked questions about the perceived usefulness of certain visual features. Specifically, we asked about the usefulness of showing the node-link diagram in the *Tree View*, the dynamic pseudocode or verbalization method in the *Information Space*, the stepping feature in the *Navigation Area*, the additional coloring to *highlight* the current node in the node-link diagram and the treemap, and the *labels* added to the nodes.

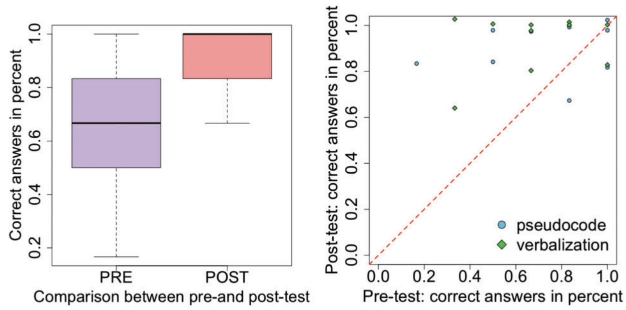

Fig. 7. *Pre-and Post-Test:* Comparison between participants' pre-test and post-test performance. (Left) Participants performed better in the post-test compared to the pre-test. (Right) Data points above the diagonal indicate participants who performed better in the post-test compared to the pre-test. Shape and color encode the two different experiment conditions. A small jitter was introduced to avoid overlapping data points.

For the overall performance, post-test scores ($MD = 92.5$ %, $n = 20$) were statistically significantly higher ($z = -3.026$, $p < .003$) compared to pre-test scores ($MD = 75$ %, $n = 20$). The effect size $r = \frac{z}{\sqrt{n}} = 0.67$ indicates a strong effect.

For comparing the two experiment conditions (*VB* and *PC*), we calculated the gain between the pre-test and the post-test. Since the often used normalized gain score ($\frac{post-pre}{1-pre}$) from Hake [85] has problems modeling a perfect pre-test result (100 % correct answers) and a possible loss (post-test performance $<$ pre-test performance), we refer to the normalized change introduced by Marx and Cummings [86] as our performance measure (1).

$$c = \begin{cases} \frac{post-pre}{1-pre} & post > pre \\ drop & post = pre = (1|0) \\ 0 & post = pre \\ \frac{post-pre}{pre} & post < pre \end{cases} . \tag{1}$$

There was no statistically significant difference between the two experiment conditions (i.e., *PC*, *VB*). The median of the normalized change for *PC* was 90 % and for *VB* 100 %.

When analyzing the final questionnaires, we found no significant differences between the two experiment groups (i.e., *PC*, *VB*). Therefore, we report the combined results in Table II.

We found a significant difference between the perceived usefulness of the explanation method in the information space (*PC* and *VB* combined) and the labels added to the node-link diagram and the treemap. Participants rated the labels ($MD = 5$, $n = 20$) higher ($z = -3.19$, $p < .002$) than the information space ($MD = 3$, $n = 19$). The effect size $r = \frac{z}{\sqrt{n}} = 0.71$ indicates a strong effect. There were no significant differences between the other visual features.

TABLE II

Feature Usefulness: In the Final Questionnaire, Participants Rated the Usefulness of the Different Visual Aids Experienced During Training

Feature	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
Tree View	5 %	0 %	20 %	30 %	45 %
Information Space	26 %	21 %	26 %	5 %	21 %
Navigation Area	15 %	10 %	0 %	25 %	50 %
Highlight	10 %	10 %	5 %	25 %	50 %
Labels	0 %	0 %	5 %	35 %	60 %

TABLE III

Motivational Predictors: Subjective Rating of Participants' Motivation During the Training Phase When Working With *TreEducation*

Predictor	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
Enjoyment	0 %	9 %	27 %	54 %	9 %
Effort	0 %	0 %	18 %	36 %	45 %
Pressure	45 %	0 %	18 %	18 %	18 %
Competence	9 %	27 %	9 %	45 %	9 %
Prefer VB	9 %	18 %	9 %	27 %	36 %

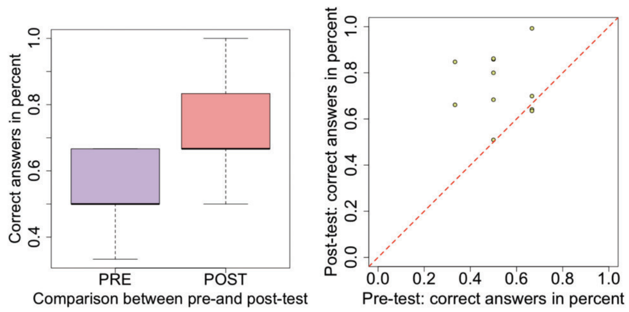

Fig. 8. *Pre-and Post-Test:* Comparison between participants' pre-test and post-test performance. (Left) Participants performed better in the post-test compared to the pre-test. (Right) Data points above the diagonal indicate participants who performed better in the post-test compared to the pre-test. A small jitter was introduced to avoid overlapping data points.

Second Session: Participants worked with the Strip layout. During the pre-and post-test, they had to answer six different questions compared to the previous session. Each question was worth one point, with six points resulting in 100 % correct answers (Fig. 8). The final questionnaire asked participants to rate whether they used the pseudocode as an explanation method or the verbalization feature and included questions concerning the predictors of motivation (i.e., enjoyment, effort, pressure, competence). Specifically, participants had to rate how much *fun* it was working with the software and whether it was *interesting*. Additionally, we asked whether they felt *anxious* while working with the software or *competent* after training.

For the overall performance, post-test scores ($MD = 66.67$ %, $n = 11$) were statistically significantly higher ($z = -2.32$, $p < .05$) compared to pre-test scores ($MD = 50$ %, $n = 11$). The effect size $r = \frac{z}{\sqrt{n}} = 0.70$ indicates a strong effect.

The results of the final questionnaires can be found in Table III.

For the motivational predictors, the invested effort ($MD = 4$, $n = 11$) was statistically significantly higher ($z = -2.17$, $p < .05$) compared to the perceived pressure ($MD = 3$, $n = 11$). The effect size $r = \frac{z}{\sqrt{n}} = 0.49$ indicates a medium effect. There were no statistically significant differences between the other motivational predictors.

3) Discussion: When comparing pre-test with post-test results, we can see a significant increase in performance when training with *TreEducation* . This is independent of the underlying layout algorithm. Therefore, we can safely state that *TreEducation* helps learners to increase their visualization literacy skills about treemaps, confirming **H1**.

As our results indicate, there is no difference in performance between participants learning with verbal descriptions (i.e., verbalization (VB)) compared to abstract explanations (i.e., dynamic pseudocode (PC)). Therefore, we cannot confirm **H2**. However, when participants can choose between the two, the majority of learners (63 %) decide on natural language, confirming **H3**. We also received feedback to include even more information in our verbalization, like a short description of the selected algorithm.

The explanation method (*PC*, *VB*) does not influence the perceived usefulness of the visual aids. People highly appreciate the visual aids implemented in *TreEducation* . Especially the labels added to the treemap and the node-link diagram were well received. This finding supports the recommendation from Firat et al. to introduce labeled nodes for a better link between the node-link diagram and the treemap [10]. A few participants (27%) commented in the free text to include even more information in the labels, like the aspect ratio of the rectangles. We refrained from implementing this feature because of the limited space available in smaller rectangles. In a future release, we will experiment with mouse-over interaction to show more details while hovering rectangles.

When analyzing the motivational predictors, we can see that people are barely anxious (45 %) about using the software and agree that it is interesting (81 %) and fun (63 %). 54 % of the learners even feel more competent. This subjective assessment confirms our hypothesis **H4**.

B. Competition as Social Gamification Element

The user experiment was conducted online via screen sharing with a video conference system. Since *competition* as a gamification element might influence the learning process positively [87], [88], [89] and, causing pressure, negatively [90], we wanted to investigate the consequences specifically in our education software.

1) Experiment Design: We decided on a between-group setting to split our participants into two groups. Both groups worked with the same restricted *TreEducation* software for training. The control group (*CG*) got access to the competitive quiz, which only showed the number of correct and wrong answers. The experimental group (*EG*) got a slightly different version of the quiz, which contained a table showing the participants' performance and the results of all other participants. To further increase *competition* and the social pressure, we ensured that participants in *EG* knew each other (e.g., colleagues, family members, or friends).

Procedure: The experimenter welcomed participants before they entered their demographic information and previous knowledge of treemaps. After this introduction, the experimenter explained the *TreEducation* application and important concepts about treemaps and hierarchical data. Participants were free to try the education software. Only the Slice-and-Dice algorithm was made available without any visual additions like color. To increase social pressure, participants in the *EG* condition were told that their results in the following quiz were made available to all following participants. All participants could stay in the training phase until they felt competent enough to complete the quiz.

The quiz consisted of 15 questions about building treemaps with the Slice-and-Dice algorithm. Each question was worth 1 point, with 15 points corresponding to 100 % correct answers. In addition to the task, a result table was present, showing the current performance of the participant. For the *EG* condition, the table was also filled with the asserted answers of the fellow participants and their names. Results were determined by mimicking different kinds of participants. One fellow has almost all answers correct, except for a few mistakes. Therefore, it is difficult but possible to beat his/her score. The other fake participants have a varying number of correct answers to simulate different performances. Participants could redo the quiz, overwrite their previous results, or revisit the training phase.

After the quiz, a final questionnaire extrapolated from the Center of Self-Determination Theory [91] was handed out to collect information about their perceived motivation. Similar to the previous experiment, we evaluated effort, pressure, enjoyment, and competence as motivational predictors. We sampled at least five semantically similar questions, formulated slightly differently or negated, for each predictor, totaling 23 questions. All questions used a 7-point Likert scale. Furthermore, participants could append feedback in the form of free text.

Tasks: Like the first experiment, we challenged participants with *size comparison*, *topology*, *understanding*, and *construction* tasks inspired by previous visualization literacy tests [10], [80], [81]. However, we incorporated the quiz directly into *TreEducation* - participants were confronted with the tree view (Fig. 1) and the treemap view, which may visualize an intermediate construction state and questions requiring the user to explore the visualizations. Eleven of the 15 tasks instructed participants to select one of three predefined options, and four demanded them to interact with the visualizations. Some exemplary tasks include:

- *Size comparison* Click on the rectangle representing the node with the highest weight. (three tasks)
- *Topology* Click on the node (in the tree view) representing the lowest common ancestor of the two marked nodes in the treemap. (six tasks)
- *Understanding* To what depth is the data visualized in the treemap? (three tasks)

TABLE IV

Motivational Predictors: Subjective Rating of Participants' Motivation After Completing the Competitive Quiz in *TreEducation*.

Predictor	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
Enjoyment	0 %	0 %	14 %	86 %	0 %
	0 %	0 %	0 %	14 %	86 %
Effort	0 %	14 %	28 %	28 %	28 %
				14 %	86 %
Pressure	29 %	57 %	0 %	14 %	0 %
	0 %	57 %	14 %	14 %	14 %
Competence	0 %	0 %	29 %	57 %	14 %
	0 %	0 %	0 %	29 %	71 %

Gray-colored cells represent the control group's answers compared to the experimental group (white cells).

- *Construction* Click on the area in the treemap where the next node will be placed. (three tasks)

Data: The data changed between the questions but was kept simple, with a maximum hierarchy level of five and not more than 17 nodes. We designed the data for each task individually, ensuring a weight difference of at least 15 % for size comparison tasks.

Participants: We recruited 14 participants (3 female, 11 male). Seven knew each other and belonged to the same social group (family or friends). Their age ranged from 21-49 (median age 23). Four participants reported a computer science background; three were experienced with data visualization, but no one was familiar with treemaps.

Hypotheses: Our hypotheses were derived from previous research on motivation [22] and social pressure [92].

- *H1: Participants' perceived motivation is higher in EG than CG.* Research has shown that competition as a gamification element positively influences learners' motivation [22].
- *H2: Participants' in EG experience more pressure.* Leaderboards showing the performance of individuals can cause stress to participants [93]. We expect the same effect to happen, although we did not include a ranked leaderboard but a summary table of correct and wrong answers. Since participants in *EG* belong to the same social group, we expect this effect to be even stronger.
- *H3: Participants' performance in terms of correct answers is higher in EG than CG.* Since we expect participants to be more motivated in the *EG* condition, they will also perform better [94].

2) Results: The questionnaire results are summarized in Table IV and split according to experimental group. A Mann-Whitney U test reveals a significant difference between the two experiment conditions given the motivation predictors and the performance.

Participants in *EG* experienced significantly more enjoyment ($MD = 5,\ n = 14,\ z = -3.07,\ p < .005$) compared to *CG* ($MD = 4, n = 14$). The effect size $r = \frac{z}{\sqrt{n}} = -0.43$ indicates a medium effect. They also invested significantly more effort ($MD = 5,\ n = 14,\ z = -2.21,\ p < .05$) and felt more competent ($MD = 5,\ n = 14,\ z = -2.22,\ p < .05$) compared to *CG* ($MD = 4,\ n = 14$). The effect size for both tests was $r = -0.31$, indicating a medium effect. There was no significant effect for pressure between the experimental groups; however, there was a slight trend ($z = -1.6$, $p < .2$) toward participants in *EG* feeling higher pressure ($MD = 2$, $n = 14$) compared to *CG* ($MD = 2$, $n = 14$).

When analyzing the performance, we found a significant effect between *EG* and *CG*. Participants in the *EG* condition were more accurate ($Mean = 99$ %, $n = 14$, $z = -2.74$, $p < .01$) than *CG* ($Mean = 81$ %, $n = 14$). The effect size ($r = -0.38$) indicates a medium effect.

We received no qualitative feedback in free text for this evaluation.

3) Discussion: We can see a significant difference in motivational predictors when comparing the two experimental groups. Participants exposed to a competitive social group enjoyed the software more, spent more effort answering questions, and finally felt more competent. Since pressure is also a motivational predictor and is not significantly different between the two conditions, we cannot support **H2** and can only partially confirm **H1**. However, for our *TreEducation* software, this result is desirable. Even without social stress, participants perform better in the *EG* condition, confirming **H3** and seem more motivated to work with the software.

VIII. Lessons Learned

Based on previous experiences implementing education tools [72], our current *TreEducation* development, and after analyzing the results of our experiments, we are able to compile the following list of lessons learned when designing education platforms:

- *Visual aids are well received by learners:*
  Participants evaluated embedding new content into existing knowledge structures via linking & brushing, accompanying labels, or highlighting relevant aspects as beneficial.
- *Steering algorithmic processes supports individual learning behaviors:*
  Participants liked the possibility to experience the algorithm stepwise. This finding is further supported by previous research on algorithm animation [7] and education software [72].
- *Verbalization helps users to understand complex topics and is preferred over more abstract explanation methods like pseudocode:*
  Results did not show a difference in performance when exposing participants to verbalization versus pseudocode. Both methods result in significant knowledge gain during training. However, the subjective preference of participants favored natural language explanation.
- *Competition as a gamification element raises learners' motivation; however, on the other hand, there is a tendency to increase social pressure:*
  Our results are in line with already conducted experiments. Although there is a benefit to the motivation of competing participants, there is also a downside. Therefore, designers must balance the pressure on learners with the expected gain on a case-by-case basis.

IX. Conclusion and Limitations

In this paper, we introduced *TreEducation* as an education platform to improve visualization literacy for treemaps. The software was built around learning theory and supported with gamification elements. In contrast to already existing approaches, our application supports a broader range of algorithms and design variations to fit a more diverse curriculum. It is available online to facilitate sharing across institutions.

Gamification elements were introduced to motivate users to learn complex concepts and test their knowledge in the same software environment. The integration of verbalization elements as explanation support was well received by learners and preferred over abstract pseudocode descriptions. The results of our experiment underline our design decisions, showing an increase in motivation.

Our experiment did not include a comparison of *TreEducation* with traditional teaching material like slides or lecture scripts. Therefore, reasoning about differences in knowledge gain between the two conditions is currently impossible. We will repeat the same experiment in upcoming semesters with a different pool of participants training solely with traditional teaching material.

We have not investigated the usefulness of *TreEducation* for teachers and lecturers. Although our usage scenario showcases one perspective, a broader view is missing. To start with, we aim to implement the software in our classrooms to support a more interactive discussion about treemaps and distribute the software across multiple institutions to collect more diverse feedback. Due to the novelty of the software, experience reports are missing but will be collected in the near future. It will be interesting to see how different teaching methods incorporate the software.

References

[1] J. Boy, R. A. Rensink, E. Bertini, and J.-D. Fekete, “A principled way of assessing visualization literacy,” *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1963–1972, Dec. 2014.

[2] K. Börner, A. Maltese, R. N. Balliet, and J. Heimlich, “Investigating aspects of data visualization literacy using 20 information visualizations and 273 science museum visitors,” *Inf. Visual.*, vol. 15, no. 3, pp. 198–213, 2016, doi: 10.1177/1473871615594652.

[3] A. V. Maltese, J. A. Harsh, and D. Svetina, “Data visualization literacy: Investigating data interpretation along the novice–expert continuum,” *J. College Sci. Teach.*, vol. 45, no. 1, pp. 84–90, 2015. [Online]. Available: http://www.jstor.org/stable/43631889

[4] D. Kunin, “Seeing theorie,” 2016. Accessed: Aug. 29, 2023. [Online]. Available: http://students.brown.edu/seeing-theory/

[5] S. Halim, Z. C. Koh, V. B. H. Loh, and F. Halim, “Learning algorithms with unified and interactive web-based visualization,” *Olympiads Informat.*, vol. 6, pp. 53–68, 2012.

[6] H. Steven and H. Felix, “Visualgo,” 2011. Accessed: Aug. 29, 2023. [Online]. Available: https://visualgo.net/

[7] S. R. Hansen, N. H. Narayanan, and D. Schrimpsher, “Helping learners visualize and comprehend algorithms,” *Interactive Multimedia Electron. J. Comput.-Enhanced Learn.*, vol. 2, no. 1, 2000. [Online]. Available: http://imej.wfu.edu/articles/2000/1/02/index.asp

[8] K. Shatri and K. Buza, “The use of visualization in teaching and learning process for developing critical thinking of students,” *Eur. J. Social Sci. Educ. Res.*, vol. 4, no. 1, pp. 71–74, Jan. 2017. [Online]. Available: https://revistia.com/index.php/ejser/article/view/6470

[9] B. Shneiderman, “Tree visualization with tree-maps: 2-D space-filling approach,” *ACM Trans. Graph.*, vol. 11, no. 1, pp. 92–99, Jan. 1992.

[10] E. E. Firat, A. Denisova, and R. S. Laramee, "Treemap literacy: A classroom-based investigation," in *Proc. Eurographics Educ. Papers*, M. Romero and B. Sousa Santos, Eds., 2020, pp. 29–38.

[11] S. Deterding, R. Khaled, L. E. Nacke, and D. Dixon, "Gamification: Toward a definition," in *Proc. Gamification Workshop*, 2011, pp. 12–15.

[12] L. Sardi, A. Idri, and J. L. Fernández-Alemán, "A systematic review of gamification in e-Health," *J. Biomed. Informat.*, vol. 71, pp. 31–48, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046417301065

[13] G. Zichermann and J. Linder, *Game-Based Marketing: Inspire Customer Loyalty Through Rewards, Challenges, and Contests*. Hoboken, NJ, USA: Wiley, 2010.

[14] R. Sevastjanova, W. Jentner, F. Sperrle, R. Kehlbeck, J. Bernard, and M. El-assady, "QuestionComb: A gamification approach for the visual explanation of linguistic phenomena through interactive labeling," *ACM Trans. Interact. Intell. Syst.*, vol. 11, no. 3/4, Sep. 2021, Art. no. 19, doi: 10.1145/3429448.

[15] Z. Turan, Z. Avinc, K. Kara, and Y. Goktas, "Gamification and education: Achievements, cognitive loads, and views of students," *Int. J. Emerg. Technol. Learn.*, vol. 11, no. 07, pp. 64–69, Jul. 2016. [Online]. Available: https://online-journals.org/index.php/i-jet/article/view/5455

[16] K. Werbach, D. Hunter, and W. Dixon, *For the Win: How Game Thinking Can Revolutionize Your Business*, vol. 1. Philadelphia, PA, USA: Wharton Digital Press, 2012.

[17] D. Robinson and V. Bellotti, "A preliminary taxonomy of gamification elements for varying anticipated commitment," in *Proc. ACM CHI Workshop Designing Gamification: Creating Gameful Playful Experiences*, 2013, pp. 69–72.

[18] S. M. Schöbel, A. Janson, and M. Söllner, "Capturing the complexity of gamification elements: A holistic approach for analysing existing and deriving novel gamification designs," *Eur. J. Inf. Syst.*, vol. 29, no. 6, pp. 641–668, 2020.

[19] A. M. Toda et al., "Analysing gamification elements in educational environments using an existing gamification taxonomy," *Smart Learn. Environ.*, vol. 6, no. 1, pp. 1–14, 2019.

[20] A. M. Toda et al., "A taxonomy of game elements for gamification in educational contexts: Proposal and evaluation," in *Proc. IEEE 19th Int. Conf. Adv. Learn. Technol.*, 2019, pp. 84–88.

[21] M. Sailer and L. Homner, "The gamification of learning: A meta-analysis," *Educ. Psychol. Rev.*, vol. 32, no. 1, pp. 77–112, 2020, doi: 10.1007/s10648-019-09498-w.

[22] M. Sailer, J. U. Hense, S. K. Mayr, and H. Mandl, "How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction," *Comput. Hum. Behav.*, vol. 69, pp. 371–380, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S074756321630855X

[23] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, "Improving participation and learning with gamification," in *Proc. 1st Int. Conf. Gameful Des. Res. Appl.*, New York, NY, USA, 2013, pp. 10–17, doi: 10.1145/2583008.2583010.

[24] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, "Studying student differentiation in gamified education: A long-term study," *Comput. Hum. Behav.*, vol. 71, pp. 550–585, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563216306203

[25] M. H. A. Rahman, D. Ismail, A. Z. B. M. Noor, and N. S. B. M. Salleh, "Gamification elements and their impacts on teaching and learning–A review," *Int. J. Multimedia Its Appl.*, vol. 10, pp. 37–46, 2018.

[26] S. A. Aljawarneh, "Reviewing and exploring innovative ubiquitous learning tools in higher education," *J. Comput. Higher Educ.*, vol. 32, pp. 57–73, 2020.

[27] S. Subhash and E. A. Cudney, "Gamified learning in higher education: A systematic review of the literature," *Comput. Hum. Behav.*, vol. 87, pp. 192–206, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563218302541

[28] K. Salen and E. Zimmerman, *Rules of Play: Game Design Fundamentals*. Cambridge, MA, USA: MIT Press, 2010.

[29] C. S. Loh, Y. Sheng, and D. Ifenthaler, *Serious Games Analytics: Theoretical Framework*. Cham, Switzerland: Springer International Publishing, 2015, pp. 3–29.

[30] R. E. Mayer, *Computer Games for Learning: An Evidence-Based Approach*. Cambridge, MA, USA: MIT Press, Jul. 2014.

[31] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel, "Large-scale network monitoring for visual analysis of attacks," in *Proc. Int. Workshop Vis. Comput. Secur.*, J. R. Goodall, G. Conti, and K.-L. Ma, Eds. Berlin, Germany:Springer, 2008, pp. 111–118.

[32] J. Bernhardt, S. Funke, M. Hecker, and J. Siebourg, "Visualizing gene expression data via Voronoi treemaps," in *Proc. 6th Int. Symp. Voronoi Diagrams*, 2009, pp. 233–241.

[33] D. Keim, S. North, C. Panse, and J. Schneidewind, "Efficient cartogram generation: A comparison," in *Proc. IEEE Symp. Inf. Visual.*, 2002, pp. 33–36.

[34] X. Chen, H. Yang, Y. Ma, and B. Zhou, "A magic treemap cube for visualizing olympic games data," *Int. J. Virtual Reality*, vol. 9, no. 3, pp. 9–17, Jan. 2010. [Online]. Available: https://ijvr.eu/article/view/2774

[35] C. Fiedler, W. Scheibel, D. Limberger, M. Trapp, and J. Döllner, "Survey on user studies on the effectiveness of treemaps," in *Proc. 13th Int. Symp. Vis. Inf. Commun. Interaction*, New York, NY, USA, 2020, Art. no. 2, doi: 10.1145/3430036.3430054.

[36] W. Scheibel, D. Limberger, and J. Döllner, "Survey of treemap layout algorithms," in *Proc. 13th Int. Symp. Vis. Inf. Commun. Interaction*, New York, NY, USA, 2020, Art. no. 1, doi: 10.1145/3430036.3430041.

[37] W. Scheibel, M. Trapp, D. Limberger, and J. Döllner, "A taxonomy of treemap visualization techniques," in *Proc. 15th Int. Joint Conf. Comput. Vis. Imag. Comput. Graph. Theory Appl.*, A. Kerren, C. Hurter, and J. Braz, Eds. SCITEPRESS, 2020, pp. 273–280.

[38] B. B. Bederson, "PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps," in *The Craft of Information Visualization*, San Francisco, CA, USA: Morgan Kaufmann, 2003, pp. 66–75.

[39] M. Bruls, K. Huizing, and J. J. van Wijk, "Squarified treemaps," in *Proc. Data Vis.*, W. C. de Leeuw and R. van Liere, Eds. Vienna: Springer, 2000, pp. 33–42.

[40] B. Shneiderman and M. Wattenberg, "Ordered treemap layouts," in *Proc. IEEE Symp. Inf. Visual.*, 2001, pp. 73–78.

[41] M. Weskamp, "newsmap," 2010. Accessed: May 02, 2023. [Online]. Available: http://newsmap.jp/

[42] B. Shneiderman, "Treemaps for space-constrained visualization of hierarchies," 1998. Accessed: Aug. 29, 2023. [Online]. Available: https://www.cs.umd.edu/hcil/treemap-history/

[43] C. Feng, M. L. Gong, and O. Deussen, "A balanced-partitioning treemapping method for digital hierarchical dataset," *Virtual Reality Intell. Hardware*, vol. 4, no. 4, pp. 342–358, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S209657962200016X

[44] S. Tak and A. Cockburn, "Enhanced spatial stability with hilbert and moore treemaps," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 1, pp. 141–148, Jan. 2013.

[45] E. Faccin Vernier, J. L. Dihl Comba, and A. C. Telea, "A stable greedy insertion treemap algorithm for software evolution visualization," in *Proc. 31st SIBGRAPI Conf. Graph. Patterns Images*, 2018, pp. 158–165.

[46] M. D'Ambros, M. Lanza, and H. Gall, "Fractal figures: Visualizing development effort for CVS entities," in *Proc. IEEE 3rd Int. Workshop Visualizing Softw. Understanding Anal.*, 2005, pp. 1–6.

[47] M. Hao, U. Dayal, D. Keim, and T. Schreck, "Importance-driven visualization layouts for large time series data," in *Proc. IEEE Symp. Inf. Visual.*, 2005, pp. 203–210.

[48] M. Hao, U. Dayal, D. Keim, and T. Schreck, "Multi-resolution techniques for visual exploration of large time-series data," in *Proc. Eurographics/IEEE-VGTC Symp. Vis.*, 2007, pp. 27–34.

[49] M. Sondag, W. Meulemans, C. Schulz, K. Verbeek, D. Weiskopf, and B. Speckmann, "Uncertainty treemaps," in *Proc. IEEE Pacific Visual. Symp.*, 2020, pp. 111–120.

[50] N. Kong, J. Heer, and M. Agrawala, "Perceptual guidelines for creating rectangular treemaps," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 990–998, Nov./Dec. 2010.

[51] J. Van Wijk and H. Van de Wetering, "Cushion treemaps: Visualization of hierarchical information," in *Proc. IEEE Symp. Inf. Visual.*, 1999, pp. 73–78.

[52] E. E. Firat, A. Joshi, and R. S. Laramee, "Interactive visualization literacy: The state-of-the-art," *Inf. Visual.*, vol. 21, no. 3, pp. 285–310, 2022.

[53] B. Alper, N. H. Riche, F. Chevalier, J. Boy, and M. Sezgin, "Visualization literacy at elementary school," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2017, pp. 5485–5497, doi: 10.1145/3025453.3025877.

[54] F. Bishop, J. Zagermann, U. Pfeil, G. Sanderson, H. Reiterer, and U. Hinrichs, "Construct-a-vis: Exploring the free-form visualization processes of children," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 451–460, Jan. 2020.

[55] K. Sobel et al., "EduFeed: A social feed to engage preliterate children in educational activities," in *Proc. ACM Conf. Comput. Supported Cooperative Work Soc. Comput.*, New York, NY, USA, 2017, pp. 491–504, doi: 10.1145/2998181.2998231.

[56] P. Ruchikachorn and K. Mueller, “Learning visualizations by analogy: Promoting visual literacy through visualization morphing,” *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 09, pp. 1028–1044, Sep. 2015.
[57] E. E. Firat, C. Lang, B. Srinivas, I. Peng, R. S. Laramee, and A. Joshi, “A constructivism-based approach to treemap literacy in the classroom,” in *Proc. Eurographics - Educ. Papers*, A. Magana and J. Zara, Eds., 2023, pp. 9–16.
[58] F. Amini, N. Henry Riche, B. Lee, C. Hurter, and P. Irani, “Understanding data videos: Looking at narrative visualization through the cinematography lens,” in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2015, pp. 1459–1468, doi: 10.1145/2702123.2702431.
[59] Z. Wang, L. Sundin, D. Murray-Rust, and B. Bach, “Cheat sheets for data visualization techniques,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2020, pp. 1–13, doi: 10.1145/3313831.3376271.
[60] W.-H. Wu, H.-C. Hsiao, P.-L. Wu, C.-H. Lin, and S.-H. Huang, “Investigating the learning-theory foundations of game-based learning: A meta-analysis,” *J. Comput. Assist. Learn.*, vol. 28, no. 3, pp. 265–279, 2012.
[61] J. Biggs and C. Tang, *Teaching for Quality Learning at University*. New York, NY, USA: SRHE and Open University Press Imprint. McGraw-Hill Education, 2011.
[62] C. M. Reigeluth, *Instructional Design Theories and Models: An Overview of Their Current Status*. Evanston, IL, USA: Routledge, 1983.
[63] J. Lave and E. Wenger, *Situated Learning: Legitimate Peripheral Participation*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
[64] S. Fox, “Situated learning theory versus traditional cognitive learning theory: Why management education should not ignore management learning,” *Syst. Pract.*, vol. 10, no. 6, pp. 727–747, 1997, doi: 10.1007/BF02557922.
[65] A. Scharle and A. Szabó, *Learner Autonomy*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[66] D. Alt, “Assessing the contribution of a constructivist learning environment to academic self-efficacy in higher education,” *Learn. Environ. Res.*, vol. 18, no. 1, pp. 47–67, 2015, doi: 10.1007/s10984-015-9174-5.
[67] T. M. Alqahtani, F. D. Yusop, and S. H. Halili, “Content validity of the constructivist learning in higher education settings (CLHES) scale in the context of the flipped classroom in higher education,” *Humanities Social Sci. Commun.*, vol. 10, no. 1, 2023, Art. no. 268, doi: 10.1057/s41599-023-01754-3.
[68] M. Wosnitza and S. Beltman, “Learning and motivation in multiple contexts: The development of a heuristic framework,” *Eur. J. Psychol. Educ.*, vol. 27, pp. 177–193, 2012.
[69] M. Bostock, V. Ogievetsky, and J. Heer, “D^3 data-driven documents,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.
[70] P. Saraiya, C. A. Shaffer, D. S. McCrickard, and C. North, “Effective features of algorithm visualizations,” in *Proc. 35th SIGCSE Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, 2004, pp. 382–386, doi: 10.1145/971300.971432.
[71] T. Barlow and P. Neville, “A comparison of 2-D visualizations of hierarchies,” in *Proc. IEEE Symp. Inf. Visual.*, 2001, pp. 131–138.
[72] J. Fuchs, P. Isenberg, A. Bezerianos, M. Miller, and D. Keim, “EduClust - A visualization application for teaching clustering algorithms,” in *Proc. Eurographics Educ. Papers*, M. Tarini and E. Galin, Eds., 2019, pp. 9–16.
[73] U. Ehsan, B. Harrison, L. Chan, and M. O. Riedl, “Rationalization: A neural machine translation approach to generating natural language explanations,” in *Proc. AAAI/ACM Conf. AI Ethics Soc.*, New York, NY, USA, 2018, pp. 81–87, doi: 10.1145/3278721.3278736.
[74] F. Hohman, A. Srinivasan, and S. M. Drucker, “TELEGAM: Combining visualization and verbalization for interpretable machine learning,” in *Proc. IEEE Visual. Conf.*, 2019, pp. 151–155.
[75] N. Feldhus, L. Hennig, M. D. Nasert, C. Ebert, R. Schwarzenberg, and S. Möller, “Saliency map verbalization: Comparing feature importance representations from model-free and instruction-based methods,” in *Proc. 1st Workshop Natural Lang. Reasoning Structured Explanations*, Toronto, Canada:Association for Computational Linguistics, 2023, pp. 30–46. [Online]. Available: https://aclanthology.org/2023.nlrse-1.4
[76] A. Jacovi and Y. Goldberg, “Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness?,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4198–4205. [Online]. Available: https://aclanthology.org/2020.acl-main.386
[77] U. Ehsan and M. O. Riedl, “Human-centered explainable AI: Towards a reflective sociotechnical approach,” in *Proc. Int. Conf. Hum.-Comput. Interaction*, C. Stephanidis, M. Kurosu, H. Degen, and L. Reinerman-Jones, Eds. Cham:Springer International Publishing, 2020, pp. 449–466.
[78] M. Linden and M. Hautzinger, *Verhaltenstherapie: Techniken, Einzelverfahren und Behandlungsanleitungen*. Berlin, Germany: Springer-Verlag, 2013.
[79] B. C. Kwon and B. Lee, “A comparative evaluation on online learning approaches using parallel coordinate visualization,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2016, pp. 993–997, doi: 10.1145/2858036.2858101.
[80] S. Lee, S.-H. Kim, and B. C. Kwon, “VLAT: Development of a visualization literacy assessment test,” *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 551–560, Jan. 2017.
[81] S. Pandey and A. Ottley, “Mini-VLAT: A short and effective measure of visualization literacy,” *Comput. Graph. Forum*, vol. 42, no. 3, pp. 1–11, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14809
[82] S. Smith-Robbins, “This game sucks: How to improve the gamification of education,” *EDUCAUSE Rev.*, vol. 46, no. 1, pp. 58–59, 2011.
[83] B. Johnson and B. Shneiderman, “Tree-maps: A space-filling approach to the visualization of hierarchical information structures,” in *Proc. Visual.*, 1991, pp. 284–291.
[84] M. E. Ahlum-Heath and F. J. Di Vesta, “The effect of conscious controlled verbalization cognitive strategy on transfer in problem solving,” *Memory Cogn.*, vol. 14, no. 3, pp. 281–285, 1986, doi: 10.3758/BF03197704.
[85] R. R. Hake, “Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses,” *Amer. J. Phys.*, vol. 66, no. 1, pp. 64–74, Jan. 1998, doi: 10.1119/1.18809.
[86] J. D. Marx and K. Cummings, “Normalized change,” *Amer. J. Phys.*, vol. 75, no. 1, pp. 87–91, Jan. 2007, doi: 10.1119/1.2372468.
[87] N. A. Mokadam et al., “Gamification in thoracic surgical education: Using competition to fuel performance,” *J. Thoracic Cardiovasc. Surg.*, vol. 150, no. 5, pp. 1052–1058, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022522315012829
[88] R. N. Landers, A. B. Collmus, and H. Williams, “The greatest battle is within ourselves: An experiment on the effects of competition alone on task performance,” *Int. J. Hum.- Comput. Stud.*, vol. 127, pp. 51–61, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581918305627
[89] L. Rodrigues, A. Toda, W. Oliveira, P. Palomino, and S. Isotani, “Just beat it: Exploring the influences of competition and task-related factors in gamified learning environments,” in *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, Porto Alegre, RS, Brasil: SBC, 2020, pp. 461–470. [Online]. Available: https://sol.sbc.org.br/index.php/sbie/article/view/12802
[90] L. Schwabe, M. Joëls, B. Roozendaal, O. T. Wolf, and M. S. Oitzl, “Stress effects on memory: An update and integration,” *Neurosci. Biobehavioral Rev.*, vol. 36, no. 7, pp. 1740–1749, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0149763411001370
[91] C. for Self-Determination Theory, “Intrinsic motivation inventory,” 2024. Accessed: Aug. 25, 2023. [Online]. Available: https://selfdeterminationtheory.org/intrinsic-motivation-inventory/
[92] J. C. Burguillo, “Using game theory and competition-based learning to stimulate student motivation and performance,” *Comput. Educ.*, vol. 55, no. 2, pp. 566–575, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131510000527
[93] C. Almeida, M. Kalinowski, A. Uchôa, and B. Feijó, “Negative effects of gamification in education software: Systematic mapping and practitioner perceptions,” *Inf. Softw. Technol.*, vol. 156, 2023, Art. no. 107142. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584922002518
[94] S. Zlate and G. Cucui, “Motivation and performance in higher education,” *Procedia - Soc. Behav. Sci.*, vol. 180, pp. 468–476, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877042815014925

Johannes Fuchs received the PhD degree in computer science from the University of Konstanz, in 2015. He is a research scientist and lecturer with the University of Konstanz, Germany. His main research interests are in the area of visual analytics and information visualization, with a special focus on data glyph design, education, gamification in higher education, game-based learning, and evaluation methods.

Bastian Jäckl received the MSc degree in computer science from the University of Konstanz, in 2023. He is a research associate with the Data Analysis and Visualization Research Group, University of Konstanz, Germany. His research focuses on Explainable AI and the exploration of graph neural networks.

Michael Jüttler received the PhD degree in business and economics education from the University of Konstanz, in 2020. He is a research associate with the Center for Innovative Teaching and Learning, School of Management and Law, Zurich University of Applied Sciences (ZHAW), Switzerland. In addition, he is a guest researcher with the University of Konstanz, Germany. He is interested in participation in teaching-learning arrangements, gamification in higher education, game-based learning, the effects of economic competencies on transitions from school to university, post-school choices, academic success, and the application of quantitative methods.

Daniel A. Keim received the PhD degree in computer science from the University of Munich, Germany. He is a professor and head of the Data Analysis and Visualization Research Group with the Computer Science Department, University of Konstanz, Germany. He has been actively involved in data analysis and information visualization research for more than 25 years and developed novel visual analysis techniques for large data sets. Before joining the University of Konstanz, he was an associate professor with the University of Halle, Germany, and Technology Consultant with AT&T Shannon Research Labs, New Jersey.

Rita Sevastjanova received the MSc degree in information engineering from the University of Konstanz, in 2017. She is a research associate with the Data Analysis and Visualization Research Group, University of Konstanz, Germany. Her research focuses on natural language processing, language modeling, and the integration of game elements into visual analytics processes to strengthen user motivation.