A Constructivism-based Approach to Treemap Literacy in the Classroom

Elif E. Firat[†] [1] Colm Lang[2] Bhumika Srinivas[2] Ilena Peng[2] Robert S. Laramee[3] Alark Joshi[2]

[1]Department of Computer Engineering, Cukurova University, Turkey
[2]Visualization and Graphics Lab, Department of Computer Science, University of San Francisco, CA, USA
[3] School of Computer Science, University of Nottingham, UK

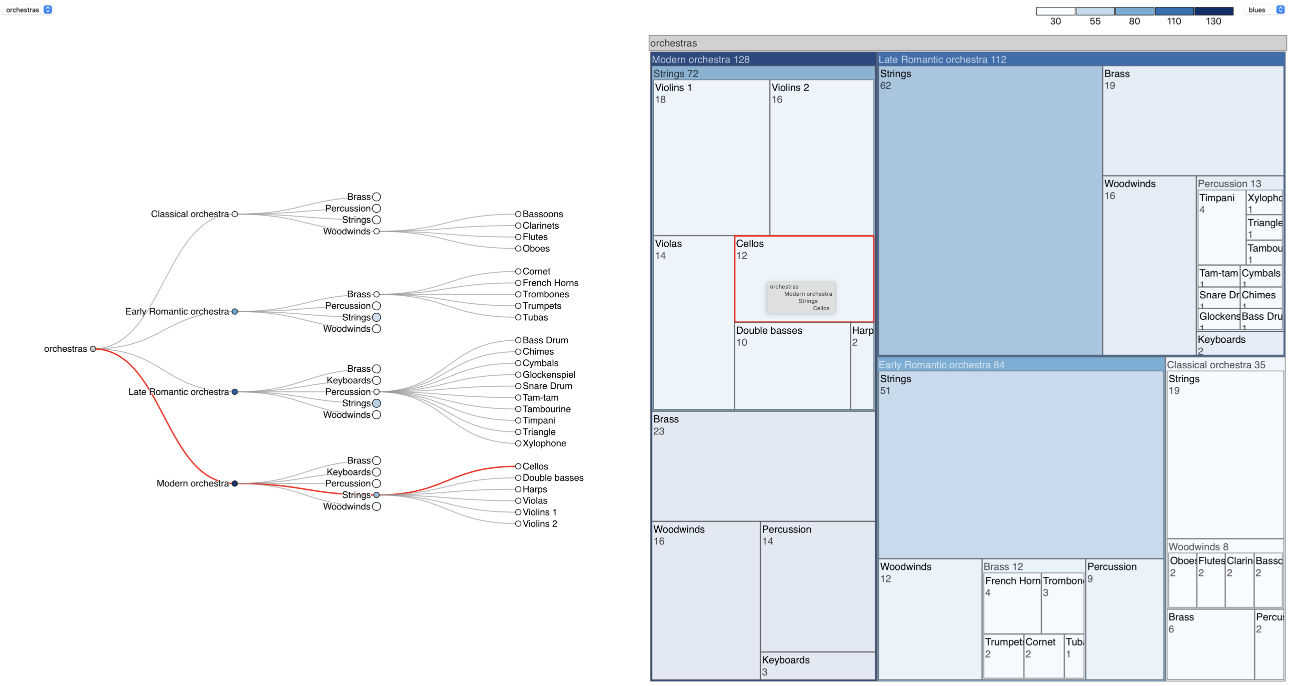


Figure 1: *This figure shows the developed interface of the interactive software for teaching students about Treemaps.*

Abstract

Treemaps are a popular representation to show hierarchical as well as part-to-whole relationships in data. While most students are aware of node-link representations / network diagrams based on their K-12 education, treemaps are often a novel representation to them. We present our experience of developing a software using principles from constructivism to help students understand treemaps using linked, side-by-side views of a node-link diagram and a treemap of the same data. Based on the qualitative survey conducted at the end of the intervention, students found the linked views to be beneficial for understanding hierarchical representation of data using treemaps.

1. Introduction

Treemaps is a popular technique that has emerged from the data visualization research community for representing hierarchical data [Shn92]. Popular news sites have been using Treemaps to communicate part-to-whole relationships such as the spending bill for a government in the New York Times [SKP21], the number of abortions in each state by the Washington Post [Ste22], and so on [PP22].

† eliffirat@gmail.com

DOI: 10.2312/eged.20231016

In this paper, we present the results of using a *Constructivism*-based approach to advancing student knowledge of Treemaps through the use of newly developed software that connects a familiar node-link representation of hierarchical data with the Treemaps representation of the same.

Constructivism is a learning theory based on the idea that people actively "construct" their own knowledge building on their prior knowledge and experiences as a foundation. The background and previous knowledge of an individual impacts how they are able to understand a specific topic. It addresses the fact that learning is an active process and is facilitated when learners actively engage with the content. To further improve the learning experience, constructivism discusses the need for educators to contextualize the subject to enable learners to better connect the new knowledge with their current foundation.

Students study network diagrams and matrix-based representations to store and manipulate data at the high school level [NYS21]. Topics include constructing a network diagram for a given problem, constructing an adjacency-matrix representation of a network, understanding the two types of representations to answer questions that require understanding the indegree/outdegree of nodes, and deriving equations based on those diagrammatic representations.

Node-link diagrams [SSKB14] are popular visual designs and their associated literacy [ZMUB18] has been studied in the data visualization community as well. Zoss *et al.* [ZMUB18] defined network visualization literacy (NVL) as the ability to read, interpret, and visualize various types of networks. They discuss a variety of topics such as how to evaluate NVL, the role of NVL in teaching and learning. Several previous studies compared node-link diagram and matrices on different types of networks with a selection of given tasks [GEY12, KEC06, OJK18, GFC05, ZMC05]. The popularity of the node-link diagram can motivate educators to use the prior knowledge of node-link diagrams as a starting point when comparing to other techniques for visualizing networks. Zhao et al. [ZMC05] specifically combined Treemaps and node-link diagrams to create a hybrid visualization technique for visualizing hierarchical data.

In an effort to increase viewer engagement, Treemaps are being used by news organizations as well as digital storytellers as a technique to convey part-to-whole relationships [SKP21]. In some cases, hierarchical data too is communicated through a static [PP22] or an interactive treemap. One of the challenges with visualizing new techniques for diverse audiences is making sense [LKH*15] of a previously unknown chart due to lack of visualization literacy. Given the prevalence of Treemaps, it is crucial that the masses have a strong grasp on how to read and interpret Treemaps.

In this paper, we present new software that was developed using principles from *Constructivism* to help students develop an understanding of Treemaps by building on their prior knowledge of node-link diagrams. We conducted a study with 38 students and found that the students provided positive feedback of the software based on the analysis of the interview that was conducted at the end of the study.

1.1. Terminology

We make a distinction between **Constructivism** as an epistemology and **constructionist** strategies situated within that epistemology, as observed by Papert and Harel [PH91]. **Constructivism** assumes learners are individuals who derive meaning from the world based on their *existing knowledge* and their personal experiences, **Constructionism** represents the possibility for "facilitating meaning-making" for those individuals through *active learning* and the process of building a shareable result [PH91].

2. Related Work

The related work section is organized into three subsections in increasing order of specificity. We first draw attention to the relevant work in the field of visualization literacy, which is followed by constructivism-based work in the field of data visualization literacy. We then provide context through the work that explores Treemap literacy.

2.1. Visualization Literacy / Pedagogy

Data visualization literacy (DVL) [BBG19a, BBG19b] can be broadly defined as the ability to read, make, and explain data visualizations. DVL therefore is not just the ability to recognize and read a new visualization technique, but to also be able to apply novel techniques for exploration, communication, and explanation of data. Prior surveys examine and classify prior research on visualization literacy and provide comparisons of related literature [FJL22a, FJL22b].

As Börner pointed out in her keynote on DVL at IEEE VIS 2019 [Bör19], it is our responsibility as data visualization experts to increase the visualization literacy not only of experts but also of general audiences [ARC*17, HNGC20].

As we aim to increase the visualization literacy of general audiences, we need softwares/tests [BRBF14, LKK16] that can be used to evaluate the current level of visualization literacy for an individual. Such community-approved tests can be used by researchers to identify baseline visualization literacy for their participants. *Treemaps* was included as one of the 12 techniques in the VLAT paper [LKK16]. Previous work in the data visualization literacy has focused on experimenting with different learning approaches to teach participants about a specific technique [KL16, PFLJ22]. In studies such as those, we compare the different modalities being used to increase visualization literacy, but we rarely evaluate whether those modalities actually made an impact on the overall visualization literacy of an individual participant or a group of participants. We also need to increase our understanding of how cognitive characteristics are related to visualization literacy so that we can enhance visualization literacy of all [LKY*19].

Rodrigues et al. [RBLB21] showed participants 20 different types of visualizations and asked them to pose as many questions as possible for each of those visualizations. Based on the analysis and classification of the user-provided questions, they discovered common mistakes that viewers make when using certain types of charts. This provides an insight into the kinds of misunderstandings that viewers may have when interpreting a chart using a certain visualization technique.

Visualization Literacy for children has been a topic of interest as well [ARC*17, HNGC20, HNGC21]. Recently, Bae et al. [BVY*22] created a physicalization-based approach where children use an assembled toolkit and a smartphone to tinker and play with visual representations. The goal was to introduce children to common visualization techniques (bar chart, line chart, pie chart) through a constructionist approach.

2.2. Constructivism

Constructivism describes the way that a student can make sense of the new material and also how the materials can be taught effectively. With Constructivism as an educational theory, the teachers consider what students *know a priori* and enable their students to put their knowledge into practice. Wojton et al. [WHHB18] introduced a Simplicity Familiarity Matrix, a study-driven paradigm for incorporating complex data visualizations into exhibition design, to ensure that all museum visitors can comprehend the images. The process of producing a data visualization was utilized to investigate which components of the data visualization are easier or harder for viewers to quickly and accurately understand.

Ruchikachorn and Mueller [RM15] propose a learning by analogy method that shows a step by step conversion between two visual designs to introduce a new visualization technique. Four visualization pair examples are used in the study to illustrate the concept. After interacting with the transitions, the participants are better able to understand the unfamiliar visual designs.

2.3. Treemap Literacy

Treemap evaluation has been a popular topic in the data visualization research field [FSL*20]. In previous work, we [FDL20] conducted an intervention that compared the use of slides and that of interactive software to teach students about treemaps. We found that interactive software was better at teaching students about treemaps than slides.

O'Handley et al. [OWW22] developed an innovative web-based platform to teach students about hierarchical data visualization techniques such as Treemaps using the *TreeVisual* system. TreeVisual focuses on one technique at a time as compared to our approach to connect a new technique with an existing technique for better understanding. Stoiber et al. [SGA22] recently presented a comparison of concrete and abstract messages when teaching individuals about Treemaps.

3. Approach

In this paper, we evaluated the benefits of using an interactive software in the intervention on student's understanding of concepts related to Treemap interpretation.

3.1. Interactive Software

To help our students understand Treemaps, we built an interactive, online application that allows them to synchronously examine node-link representation of the data along with a treemap representation of the data. The software can be found at `https://nodelink-treemap.github.io/`.

Figure 1 shows the interface of the program, where the path to a specific node in the hierarchy is highlighted in red in both representations (the node-link representation on the left and the treemap on the right). In the treemap view, we also show the viewer a tooltip that shows the hierarchy of the current node (which in this case is $orchestras \rightarrow Modern\ orchestra \rightarrow Strings \rightarrow Cellos$).

Synchronized Interaction is an important design goal for the team to follow Constructivism principles, due to the fact that most students are well versed with network/node-link diagrams. The red line in Figure 1 shows the interaction that a viewer experiences when hovering over the corresponding "Cellos" node in the treemap.

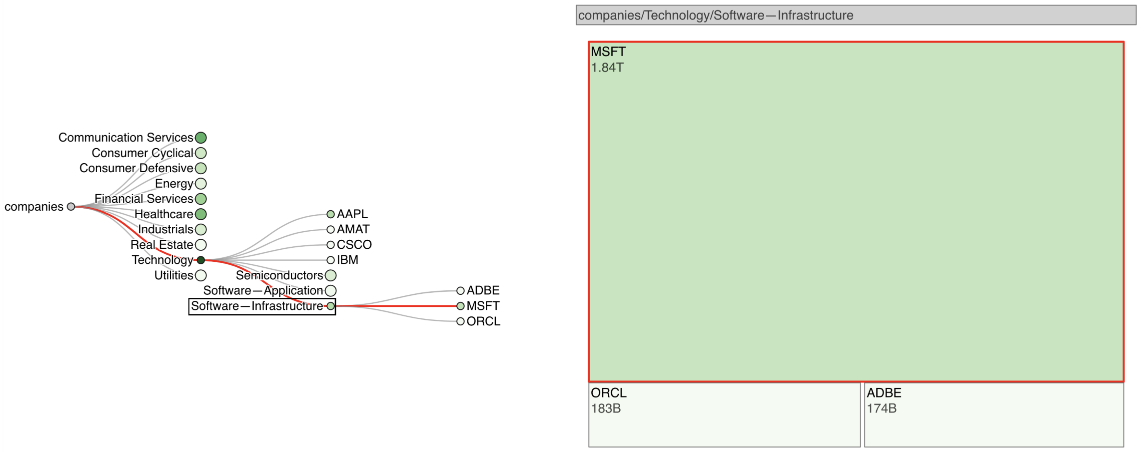
A student can interact with the treemap by hovering to examine a node or by clicking on a node to zoom into its sub-tree. Figure 2 shows a series of screenshots where a student has drilled all the way down to a leaf node and is then navigating back up the tree using the navigation bar on top of the treemap. The immediate parent of the current node is shown using a rectangular outline on the node link diagram, and the path to the current node of interest is colored in red to help the viewer gain a better understanding of how enclosure works in treemaps to represent hierarchical data.

The *Color Legend* is displayed in the top right corner of the software to provide a visual cue about the number of elements contained in a node. Figure 3 shows an example where a node (Technology) is minimized in the left snapshot and expanded in the right snapshot. As the *Technology* sector has a high aggregate value, the *Technology* rectangle and corresponding circle in the node-link diagram are colored dark purple. On expanding the *Technology* node in the node-link diagram, the contained nodes are colored based on their aggregated/individual values (as shown in the bottom snapshot of Figure 3).

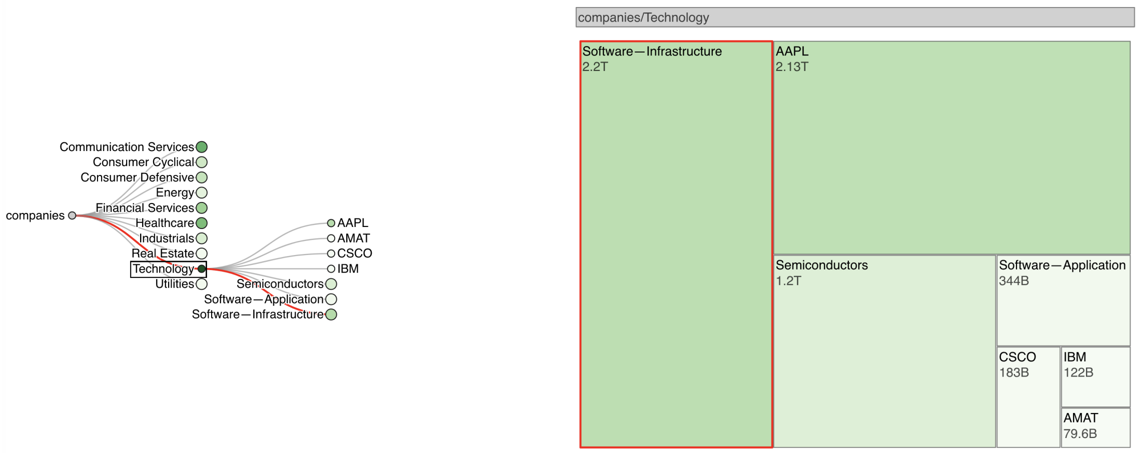
3.2. Implementation Details

The interactive software was built using React [Met23] and D3 [BOH11]. We used React due to its ability to perform state handling. We developed reusable visualization components that provide simplicity through handling getter/setter functions to trigger a re-render. Because of the reusable charts paradigm, there is a necessity to maintain idempotency. Oftentimes, if we need to simply append an element, the .append() method will do that for every render. This is not what we want, so the solution we use here is to use .data([null]) and .join() so that the "one time element" is bound to an arbitrary singular null data point. This ensures that every render will either add or update, instead of rendering multiple times on top of one another. On a similar note, because the my() function runs in each chart at each render, we used the general update pattern with .join(enter, update, exit) to maintain transition logic. This was especially useful when handling a collapsed node's *exiting children* and an expanded node's *entering children*.

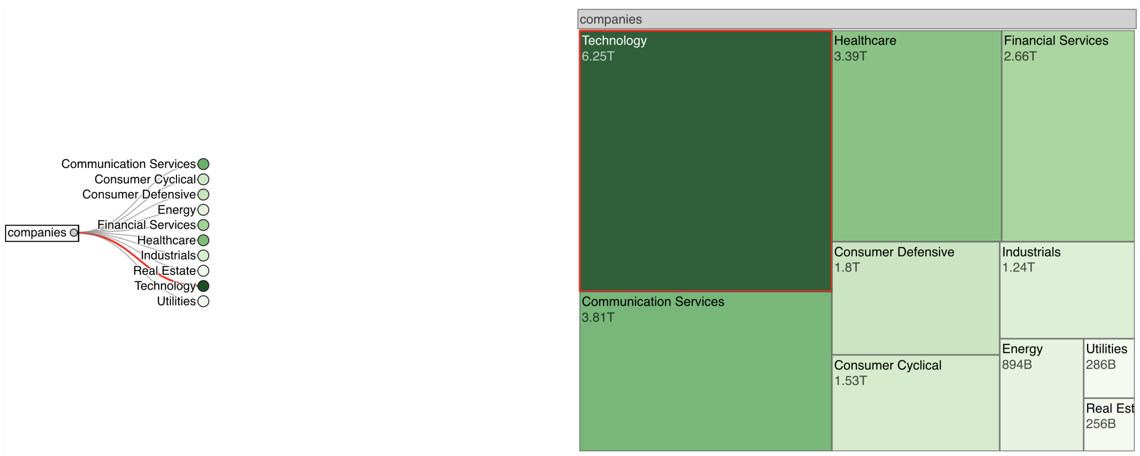
The my() function also facilitates D3-style chaining. When the nodelink().data(...) is invoked, the data setter function returns *my* after setting data. This facilitates the D3 chaining. It is only used to trigger renders and to serve as the data to return for chaining. This allows the chart function to form a closure that accepts a selection. The entire function chain for the chart eventually returns



(a) *Zoomed into the lowest level in the hierarchy. Clicking on the Treemaps navigation bar (companies→ Technology→ Software - Infrastructure.) zooms out one level to show Figure 2b below.*



(b) *Zoomed out one level. Clicking on the Treemaps navigation bar (Companies→ Technology) zooms out one more level to show Figure 2c below.*



(c) *Zoomed out one more level to show the first level of nodes in the hierarchy.*

Figure 2: *The figure above shows snapshots of the various levels of zooming that are possible by interacting with the Treemap software. Clicking on a rectangle in the Treemap allows zooming in, whereas clicking on the navigation bar on top of the Treemap allows the student to zoom out one level at a time.*

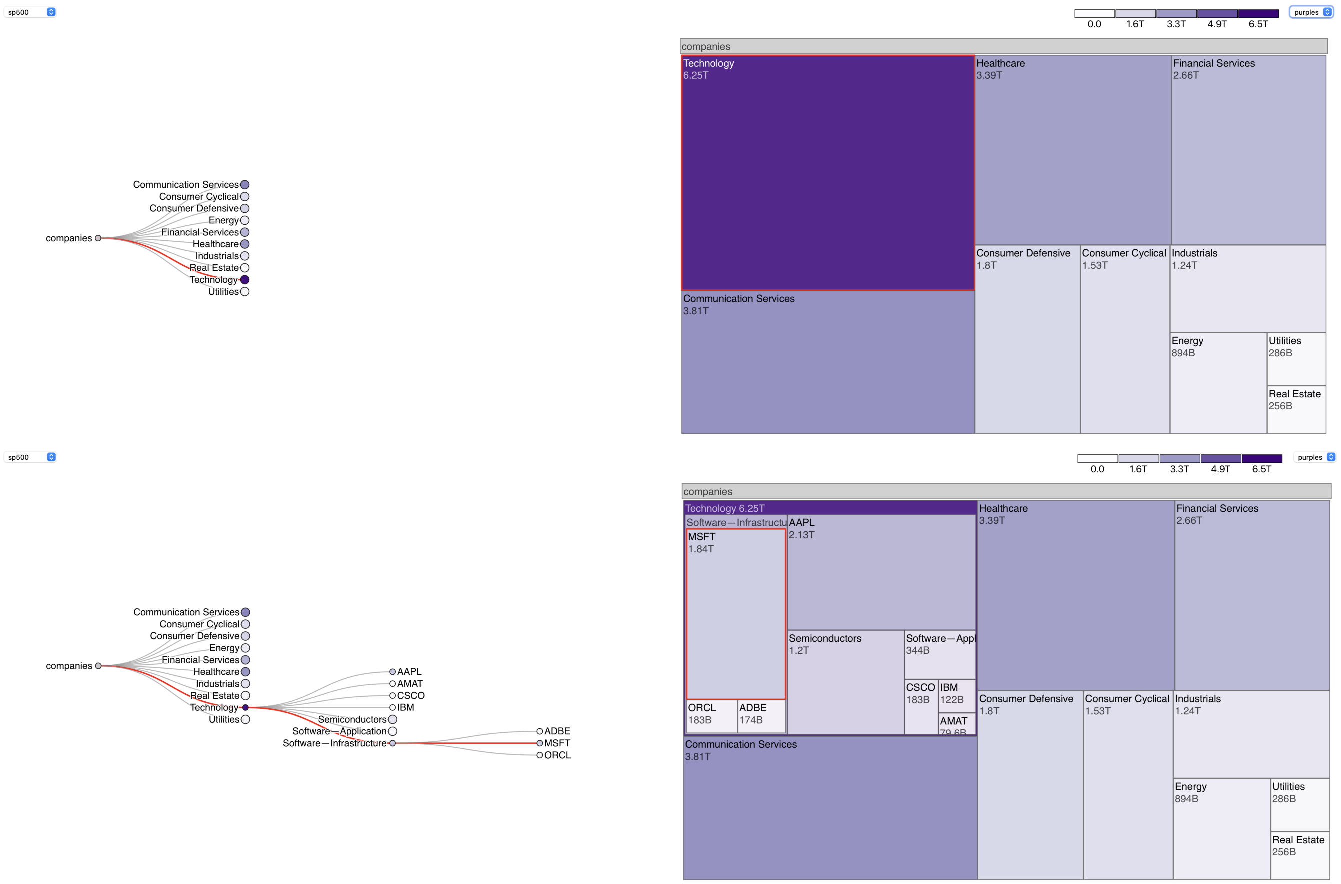


Figure 3: *The two screenshots above show a snapshot of the before and after a node has been expanded and the way the color map is used to convey the numeracy of a node. The top figure shows the Technology rectangle (top left) colored in dark purple, whereas on the right the Technology rectangle is further subdivided into its various children, where each node is also colored based on the color map shown on the top right.*

my(selection) so that when svg.call(nodelink) is called, it actually invokes my(svg).

The Treemap and node-link share the underlying hierarchical data, including its current state of collapses/expanded nodes. We sort the nodelink nodes alphabetically and the treemap by value.

In this software, we allow users to explore four different datasets - (i) an S&P 500 dataset, (ii) the Flare dataset [Bos23], (iii) a dataset about various instruments in an Orchestra [Wik23], and (iv) a dataset about the Classification of Animals.

The interactive software can be found online at `https://nodelink-treemap.github.io/` and source code can be found at `https://github.com/nodelink-treemap/nodelink-treemap.github.io`. We also provide a supplementary video that demonstrates the features of the software.

3.3. Procedure

We conducted an intervention to teach students about Treemaps in two separate Data Visualization courses at the University of San Francisco. The procedure for our educational intervention is as follows:

1. Students fill out a demographics survey
2. Students perform common tasks (expand, compare, hover, etc.) as they interact with the Treemaps software (see Section 3.4)
3. Student share their experience with the software by filling out a qualitative survey

The first data visualization course had 28 students (22 undergraduate students, 6 graduate students). Out of those, 18 students were in the 18-22 age group, 8 students in the 23-27 age group, and 2 students in the 33-37 (7 females and 21 males).

For a subsequent data visualization course, 10 more students (5 undergraduate students, 5 graduate students) used the software. 7 students were in the 18-22 age group, 2 students in the 23-27 age group, and 1 in the 28-32 age group (3 females and 7 males). We expanded the questionnaire for these 10 students (see Figure 4) to record their previous exposure to node-link diagrams. The two new questions are highlighted in the list of questions provided in Section 3.5. 9 out of 10 students rated their familiarity at either 5 or 6

Proceedings published by Eurographics - The European Association for Computer Graphics.

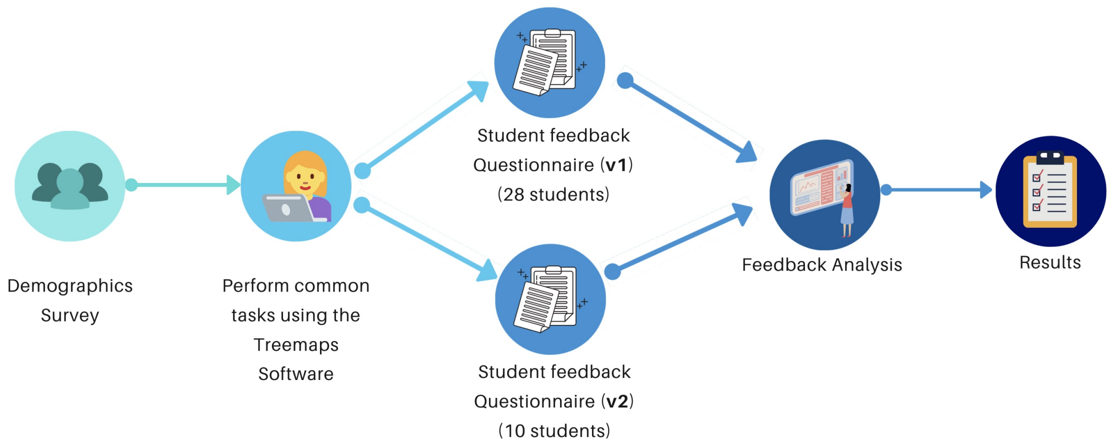


Figure 4: *Procedure followed for the pedagogical intervention. After the demographics survey, students were asked to perform common tasks using the treemaps software. Student feedback was then collected using a questionnaire. Two more questions were added to the questionnaire (v2) for the second offering of the data visualization course. We analyzed the student feedback to identify the impact of the software on student understanding of treemaps.*

on a scale of 1 (not at all familiar) to 7 (extremely familiar), while one student rated their familiarity at 2.

3.4. Tasks

Here are the tasks that we asked all our students to perform when experimenting with the software:

- Find the corresponding node in the treemap (see Figure 1)
- Locate the path of a node in the node-link diagram and the treemap (see Figure 1)
- Identify the node that has the greatest number of children in the treemap (see Figure 2c)
- Use the color scale to compare two nodes in the treemap (see Figure 3)
- Compare the length of the path of two nodes using the node-link diagram and the treemap (see Figure 1)

3.5. Post-Intervention Interview Questions

Here are the questions from the developed questionnaire that we used to solicit feedback from the students about the impact of the software on their understanding. 38 total students answered the questions listed below. Questions highlighted in bold were answered only by the students in the second offering of the data visualization course.

1. Have you seen Treemaps before? If yes, where?
2. **Have you seen network / node-link diagrams before? If yes, where?**
3. **Please rate your familiarity with network / node-link diagrams on a scale of 1-7.**
4. Do you have a background in Data Visualization? If so, what is it?
5. Did the Treemaps software help you gain a better understanding of Treemaps? (Yes/No)
6. Why (or why not) do you think it was helpful?
7. Rate the efficacy of Treemaps software to visualize hierarchical data on a scale of 1 (not at all) - 7 (very much).
8. Why (or why not) do you think it was effective?
9. Do you recommend any improvements to the software?
10. What was your understanding of Treemaps **before** you interacted with the Treemaps software? Rank on a scale of 1 (not at all) to 7 (very much).
11. What was your understanding of Treemaps **after** you interacted with the Treemaps software? Rank on a scale of 1 (not at all) to 7 (very much).

4. Results

In order to evaluate the students' experience with our interactive treemap software, we analyzed their answers to post-software interview questions provided at the end of the study (see Section 3.5.

While 23 of the students (61%) reported they had never seen a Treemap before, only 3 of them (8%) had prior experience with data visualization. Students were asked to reflect on their experience of using the software whether it helped them to gain better understanding of Treemaps. 29 students (76%) stated that the software helped them to comprehend the Treemap and that the software had positive aspects such as being useful in understanding the data hierarchy and the relationship between different data attributes. Here is some student feedback: *"It's easier to start from a larger scope, then dig deeper than giving the whole view with all the details."* (S3), *"Easy to see the relationship between different fields."* (S8), and *"...The visualization of the tree is helpful to show where in the category you are currently looking at."* (S20)

When we asked them to rate how effective our interactive Treemap software was in showing the data hierarchy, 29 students (76%) found it very effective and gave it a rating of 5 and above, while 9 of them (24%) gave it a rating of 4 or less (see Figure 5).

They were also asked to report whether they found the Treemap software effective and 9 of them (24%) found the software easy to read and understand the data: *"It guides you through the tree making it easy to know where you are in the tree"* (S18). Also, 19 students (50%) stated that it was effective because the hierarchy was shown explicitly: *"The data hierarchy is clearly represented in the software."* (S8), and *"I think it was very effective because it starts at the beginning and expands from there and shows the overall boxes and how they each get more detailed."* (S34).

6 students (16%) observed that interaction and animation make the software robust: *"It is helpful to use the interaction to see how the treemap adapts and what it is made up off. You can minimize or maximize it, and just working through it helps one get a better understanding of the overall structure."* (S33). While 10 students (26%) reported only that the treemap software was effective and intuitive, a total of 31 students (82%) stated that the software is powerful.

In addition to these questions, we asked them what they could recommend to improve the Treemap software. We received a suggestion about providing different colors: *"Color schemes could be improved."* (S6).

Figure 6 shows the overall improvement in the understanding of Treemaps as assessed by the students. The median understanding of Treemaps went from 3 (before) to 5.5 (after) using the software.

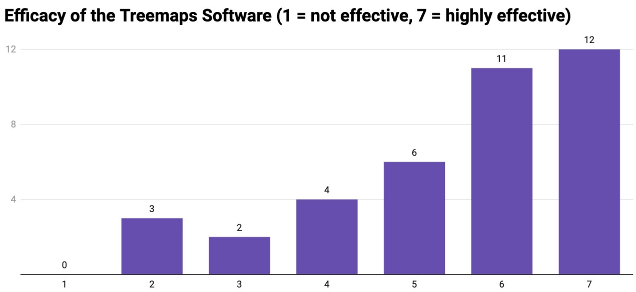


Figure 5: *Students were asked to rate the efficacy of our Treemaps software to visualize hierarchical data on a scale of 1 (not at all effective) - 7 (extremely effective). 29 students rated it as effective, highly effective, or extremely effective.*

5. Conclusion & Future Work

Treemaps are being increasingly used in the media to communicate part-to-whole or hierarchical data. In this paper, we presented our findings on a Constructivism-based approach to teaching treemaps using the node-link representation of a tree as a starting point. To enable a deeper understanding of Treemaps, we developed an application that provides bi-directional synchronous interaction to help students understand treemaps as they interacted with a node-link diagram and vice versa. Based on the deployment of the software and a subsequent evaluation, students mentioned that the interactive software helped improve their understanding of treemaps.

Future work could compare and contrast the impact of our interactive software with other modalities such as a video recording,

Figure 6: *The figure shows the overall improvement in understanding of Treemaps using the software. The median before using the software was 3, whereas the median after using the software was 5.5.*

a live lecture, or a data comic that explains how treemaps are constructed. Additionally, we also plan to evaluate the benefits in learning by conducting a pre- and post-test with our students. We can then use the results of the post-test to identify common stumbling blocks faced by the students when learning about Treemaps.

References

[ARC*17] ALPER B., RICHE N. H., CHEVALIER F., BOY J., SEZGIN M.: Visualization literacy at elementary school. In *Proceedings of the 2017 CHI conference on human factors in computing systems* (2017), pp. 5485–5497.

[BBG19a] BÖRNER K., BUECKLE A., GINDA M.: Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments. *Proceedings of the National Academy of Sciences 116*, 6 (2019), 1857–1864.

[BBG19b] BÖRNER K., BUECKLE A., GINDA M.: Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments. *Proceedings of the National Academy of Sciences 116* (2019), 1857 – 1864.

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE transactions on visualization and computer graphics 17*, 12 (2011), 2301–2309.

[Bör19] BÖRNER K.: Vis keynote address: Data visualization literacy. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2019), IEEE, pp. 1–1.

[Bos23] BOSTOCK M.: Flare library class dependency. https://github.com/d3/d3-hierarchy/blob/main/test/data/flare.json, 2023. [Online; accessed 18-January-2023].

[BRBF14] BOY J., RENSINK R. A., BERTINI E., FEKETE J.-D.: A principled way of assessing visualization literacy. *IEEE transactions on visualization and computer graphics 20*, 12 (2014), 1963–1972.

[BVY*22] BAE S. S., VANUKURU R., YANG R., GYORY P., ZHOU R., DO E. Y.-L., SZAFIR D. A.: Cultivating visualization literacy for children through curiosity and play. *IEEE Transactions on Visualization and Computer Graphics* (2022).

Proceedings published by Eurographics - The European Association for Computer Graphics.

[FDL20] FIRAT E. E., DENISOVA A., LARAMEE R. S.: Treemap literacy: A classroom-based investigation. In *Eurographics Proceedings* (2020).

[FJL22a] FIRAT E. E., JOSHI A., LARAMEE R. S.: Interactive visualization literacy: The state-of-the-art. *Information Visualization 21*, 3 (2022), 285–310.

[FJL22b] FIRAT E. E., JOSHI A., LARAMEE R. S.: Vislite: Visualization literacy and evaluation. *IEEE Computer Graphics and Applications 42*, 3 (2022), 99–107.

[FSL*20] FIEDLER C., SCHEIBEL W., LIMBERGER D., TRAPP M., DÖLLNER J.: Survey on user studies on the effectiveness of treemaps. In *Proceedings of the 13th international symposium on visual information communication and interaction* (2020), pp. 1–10.

[GEY12] GHANI S., ELMQVIST N., YI J. S.: Perception of animated node-link diagrams for dynamic graphs. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1205–1214.

[GFC05] GHONIEM M., FEKETE J.-D., CASTAGLIOLA P.: On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information visualization 4*, 2 (2005), 114–135.

[HNGC20] HUYNH E., NYHOUT A., GANEA P., CHEVALIER F.: Designing narrative-focused role-playing games for visualization literacy in young children. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 924–934.

[HNGC21] HUYNH E., NYHOUT A., GANEA P. A., CHEVALIER F.: Designing narrative-focused role-playing games for visualization literacy in young children. *IEEE Transactions on Visualization and Computer Graphics 27* (2021), 924–934.

[KEC06] KELLER R., ECKERT C. M., CLARKSON P. J.: Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? *Information Visualization 5*, 1 (2006), 62–76.

[KL16] KWON B. C., LEE B.: A comparative evaluation on online learning approaches using parallel coordinate visualization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 993–997.

[LKH*15] LEE S., KIM S.-H., HUNG Y.-H., LAM H., KANG Y.-A., YI J. S.: How do people make sense of unfamiliar visualizations?: A grounded model of novice's information visualization sensemaking. *IEEE transactions on visualization and computer graphics 22*, 1 (2015), 499–508.

[LKK16] LEE S., KIM S.-H., KWON B. C.: VLAT: Development of a visualization literacy assessment test. *IEEE transactions on visualization and computer graphics 23*, 1 (2016), 551–560.

[LKY*19] LEE S., KWON B. C., YANG J., LEE B. C., KIM S.-H.: The correlation between users' cognitive characteristics and visualization literacy. *Applied Sciences 9*, 3 (2019), 488.

[Met23] META: React - a javascript library. https://reactjs.org/, 2023. [Online; accessed 27-January-2023].

[NYS21] NEW YORK STATE: Common core curriculum - Introduction to Networks. https://studylib.net/doc/7241596/precalculus-module-2--topic-a--lesson-1--teacher, 2021. [Online; accessed 23-October-2022].

[OJK18] OKOE M., JIANU R., KOBOUROV S.: Node-link or adjacency matrices: Old question, new insights. *IEEE transactions on visualization and computer graphics 25*, 10 (2018), 2940–2952.

[OWW22] O'HANDLEY B., WU Y., WANG C.: Treevisual: Design and evaluation of a web-based visualization tool for teaching and learning tree visualization. In *2022 ASEE Annual Conference & Exposition* (2022).

[PFLJ22] PENG I., FIRAT E. E., LARAMEE R. S., JOSHI A.: Evaluating bloom's taxonomy-based learning modules for parallel coordinates literacy. *Computer Graphics Forum* (2022).

[PH91] PAPERT S., HAREL I.: Situating constructionism. *constructionism 36*, 2 (1991), 1–11.

[PP22] PALLARO B., PARLAPIANO A.: Visualizing the 13.6 billion in u.s. spending on ukraine. https://www.nytimes.com/interactive/2022/03/18/upshot/ukraine-aid-details.html, 2022. [Online; accessed 23-April-2022].

[RBLB21] RODRIGUES A. M. B., BARBOSA G. D. J., LOPES H. C. V., BARBOSA S. D. J.: What questions reveal about novices' attempts to make sense of data visualizations: Patterns and misconceptions. *Computers & Graphics 94* (2021), 32–42.

[RM15] RUCHIKACHORN P., MUELLER K.: Learning visualizations by analogy: Promoting visual literacy through visualization morphing. *IEEE Transactions on Visualization and Computer Graphics 21*, 9 (2015), 1028–1044.

[SGA22] STOIBER C., GRASSINGER F., AIGNER W.: Abstract and concrete materials: What to use for visualization onboarding for a treemap visualization? In *Proceedings of the 15th International Symposium on Visual Information Communication and Interaction* (2022), pp. 1–9.

[Shn92] SHNEIDERMAN B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG) 11*, 1 (1992), 92–99.

[SKP21] SANGER-KATZ M., PARLAPIANO A.: The democrats have a lot of cutting to do. https://www.nytimes.com/bill.html, 2021. [Online; accessed 23-April-2022].

[SSKB14] SAKET B., SIMONETTO P., KOBOUROV S., BÖRNER K.: Node, node-link, and node-link-group diagrams: An evaluation. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2231–2240.

[Ste22] STEVENS H.: Majority of women ages 15 to 44 would face new post-roe abortion limits. https://www.washingtonpost.com/nation/2022/05/04/abortion-numbers-us-roe-opinion/, 2022. [Online; accessed 23-April-2022].

[WHHB18] WOJTON M. A., HAYDE D., HEIMLICH J. E., BÖRNER K.: Begin at the beginning: A constructionist model for interpreting data visualizations. *Curator: The Museum Journal 61*, 4 (2018), 559–574.

[Wik23] WIKIPEDIA: Orchestra. https://en.wikipedia.org/w/index.php?title=Orchestra&oldid=1022472978, 2023. [Online; accessed 18-January-2023].

[ZMC05] ZHAO S., MCGUFFIN M. J., CHIGNELL M. H.: Elastic hierarchies: Combining treemaps and node-link diagrams. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (2005), IEEE, pp. 57–64.

[ZMUB18] ZOSS A., MALTESE A., UZZO S. M., BÖRNER K.: Network visualization literacy: novel approaches to measurement and instruction. In *Network Science In Education*. Springer, 2018, pp. 169–187.