electronics

MDPI

Article

An Innovative JavaScript-Based Framework for Teaching Backtracking Algorithms Interactively

Moustafa M. Nasralla

Department of Communications and Networks Engineering, Prince Sultan University, Riyadh 11586, Saudi Arabia; mnasralla@psu.edu.sa

Abstract: Algorithm fundamentals are useful to learn at different levels engineering education. One of the most difficult concepts to teach and understand is backtracking algorithms with proper bounding functions. This article proposes a framework to implement interactive online tools showing examples of backtracking algorithms in which students can graphically observe execution step-by-step. This approach is illustrated with the n-queens problem with students from Prince Sultan University, Saudi Arabia, and Complutense University of Madrid, Spain. The results show 6.67% increased learning on a backtracking exercise in the experimental group over the control group, in which the algorithms were automatically validated with DOMjudge software (an automated system used to run programming contests). The proposed framework was evaluated as easy to use, with a score of 74.5% in the validated System Usability Scale (SUS); easy to learn, with a score of 6.22 out of 7 in the validated Usefulness, Satisfaction, and Ease-of-Use (USE) scale; and with a general satisfaction of 5.97 out of 7 in the validated USE scale.

Keywords: backtracking; engineering education; digital learning; higher education; fundamentals of algorithms; programming; online application; software algorithms

Citation: Nasralla, M.M. An Innovative JavaScript-Based Framework for Teaching Backtracking Algorithms Interactively. *Electronics* **2022**, *11*, 2004. https://doi.org/10.3390/electronics11132004

Academic Editor: George Angelos Papadopoulos

Received: 24 April 2022
Accepted: 17 June 2022
Published: 27 June 2022

1. Introduction

Learning of algorithms is incorporated into different levels of engineering education such as telecommunication and computer science, and in different countries and different cultures. Algorithms are behind most automation processes covering a wide range of fields, including decision-making processes [1], path-finding in games with artificial intelligence (AI) [2], and telecommunication routing [3]. Proper design of algorithms is the cornerstone for efficient and effective applications and communications.

Although there are many different algorithms (almost as many as there are problems), there are some big sets of algorithms that fall into a few well-known categories. These algorithm techniques are usually taught at different university grades, and include iterative [4], recursive [4], divide and conquer [5], and backtracking algorithms [6], among others. In particular, backtracking is one of the most difficult to understand by students, as it combines recursion, exploration of partial solutions, and bound of branches to reach reasonable efficiency.

In higher education and collaborative learning, it is essential to adopt a teaching strategy that develops excitement and collaboration among the students. Dynamic group formation with intelligent tutor collaborative learning for next-generation collaboration was proposed in [7]. In fact, group formation strongly influences collaborative learning in computer-supported collaborative learning environments. Since group formation is affected by several factors (such as personal characteristics and social, cultural, psychological, and cognitive diversity), dynamic group formation was developed to solve concerns related to group compatibility among students with different traits. In addition, face-to-face tutoring provides a learning environment that suits student preferences and levels of understanding (i.e., learning styles and learning levels, respectively). Moreover, the intelligent

tutoring system is a computer-based system that integrates artificial intelligence to mimic the cognitive intelligence of human tutors. The authors of [8] proposed blending cognitive intelligence with their proposed intelligent tutoring system to replace existing traditional tutoring systems. Another study [9] supports increasing student responsibility and learning outcomes by using a partially flipped classroom in a language processor course. Recently, teaching pedagogy in higher-education incorporates watching videos or using interactive applications without any detailed face-to-face introduction of the topic beforehand. This indicates that most teaching frameworks will probably be useful for creating material for flipped classrooms.

Teaching algorithms in higher education require learners to be well prepared beforehand. The authors of [10] proposed dynamic optimization and hyperparameter optimization of deep neural network models to improve the accuracy of prediction model positioning and software quality and testing efficiency. The prediction model was developed by mixing the particle swarm algorithm and the wolf swarm algorithm, which takes advantage of the strengths of each algorithms. Moreover, teaching abstraction in computer science is regarded as one of the most fundamental ideas, and can be achieved by using abstraction and ignoring details that are currently irrelevant. The authors of [11] studied the effect of abstraction frameworks in a primary computer science course called Scratch. This tool uses the programming language for implementing algorithmic solutions. The outcome of the study was that the framework was highly effective for developing computer science abstraction skills as well as other related skills and aspects, such as the tendency to provide explanations for solutions, the use of initialization processes, and perception of the nature of computer science.

Our approach proposes to address this cross-disciplinary need of learning algorithm techniques by using a cross-disciplinary education technique. More concretely, cross-disciplinary education usually relies on interactive tools to effectively teach concepts due to their intuitiveness and ease-of-use [12]. The authors of [13] accurately assess the efficacy of learning and teaching techniques through analyzing learning outcome performance by using an advanced analytical model (the Rasch model). The results from this work guide educators in tracking and monitoring learning outcome performance for different types of courses, such as algorithm-based education. Hence, our proposed approach relies on a framework for creating interactive tools for teaching algorithms.

Understanding fundamentals of algorithms helps students to develop accurate solutions that actually obtain the best solutions in the most efficient way considering total time and computational complexity. However, teaching and learning this subject can be a real challenge, as mentioned in other works about teaching algorithmic thinking, such as [14]. For instance, one aspect of fundamentals of algorithms that is most difficult to teach and understand is backtracking with proper bounding functions, as it combines recursion, exploration of partial solutions, and bound of branches to reach reasonable efficiency. In this paper, we propose a framework to implement interactive online tools to show examples of backtracking algorithms in which students can graphically observe execution step-by-step. This work conducts experiments with Prince Sultan University (PSU) students and Complutense University of Madrid (UCM) students in order to show the perceptions students have of this framework and the corresponding interactive tool, and to help when these tools are reused in different contexts and countries.

The remainder of the article is organized as follows. The next section introduces related work highlighting the gap in the literature covered by this work. Section 3 presents a novel framework for developing interactive tools for teaching algorithms. Section 4 presents a case study of using this framework for teaching the backtracking algorithm for solving the n-queens problem. Section 5 presents experimentation with this interactive tool. Section 6 discusses the main findings of this research. Section 7 mentions the conclusions and depicts the most-relevant future research directions.

2. Related Work

Several works show the relevance of backtracking algorithms in different fields. For example, Zhang et al. [15] have recently presented a backtracking search algorithm for identifying certain parameters in photovoltaic models in the energy field. In addition, backtracking has been used for implementing the Root-Tracker [16] tool to identify the true source of cyber crime. Thus, engaging students to properly understand and program backtracking algorithms can contribute to the quality of future tools developed with this technique.

Gamification has been widely used in education. In particular, a variety of games have been used for teaching software process standards. The literature review by Calderon et al. [17] showed that the majority of serious games for supporting software process standards education were computer games and applicable for industrial environments. These games were mainly used by professors and undergraduate students. Thus, software development has already benefited from serious games. Our proposed work is in line with this trend, as it presents backtracking algorithms in a more visual, interactive, and playful way.

The authors of [18] described the advantage of using three-dimensional (3D) technology to support teaching and learning in healthcare education and for assessment of learning outcomes. The assessed learning outcomes were related skills, knowledge, student perceptions, and emotions. These learning outcomes were beneficial to support learning and teaching strategies for interactive education. Gamification, on the other hand, plays a vital role in education. It supports and motivates students in understanding complicated topics (such as algorithmic teaching), which can lead to enhanced learning and outcomes [19]. Moreover, it has been observed that teaching and learning of different educational specialties requires the support of computer science technologies such as visualization, pedagogical simulations, interactive tools, etc. For instance, the authors of [20] developed a simulation visualization tool to graphically illustrate various computer science concepts by simulating the network-to-memory data path at a network node and generating files for simulation visualization of hardware description languages-based models. Our proposed work also considers enhancement of learning through the development of an interactive online tool to solve problems for engineering and computer science students.

Despite the widespread availability and increasing use of cyberlearning environments, the authors of [21] evaluate a teaching framework for cyberlearning environments in computer science and software engineering education. The evaluation framework proposes development of a user interface (UI) and user experience (UX) to support teaching and learning for STEM education. Furthermore, the authors of [22] developed an educational software tool to analyze sound propagation to provide interactive understanding and learning of acoustic engineering principles. In addition, visualization of algorithms has helped different branches of software-development education. For instance, virtual reality modeling language and Java were used to develop applets for visualizing 3D algorithms in computer graphics education [23]. Our proposed work also focuses on visualizing algorithms, but the type of algorithm, backtracking, is different.

To enhance student computational thinking skills, the authors of [24] developed an immersive virtual reality-based application to facilitate learning computational thinking concepts. The goal of this study was to support traditional teaching and learning of computational thinking by integrating three virtual mini games for an enhanced learning experience. Furthermore, analysis of algorithms is essentially required as part of computing curriculum at different educational levels. In general, students find such subjects to be complicated and difficult due to their theoretical and mathematical nature. Hence, the authors of [25] designed a tool-set named "BiGO" to support computer science students in learning how to analyze time complexities of algorithms in courses such as data structures and algorithms, and design and analysis of algorithms.. In addition, the pedagogical goals of the tool and the system implementation architecture were outlined. Another study [26] designed a smart learning environment to facilitate computational thinking education in Nigeria. Smart learning needs to align well with the requirements of smart cities [27].

Therefore, smart learning is a new learning approach, where learner-centered pedagogy and advanced technology play major roles in education. Computational thinking is a requisite for students in the digital age to prepare them for future career challenges. These computational thinking skills allow students to solve problem decomposition, abstraction, algorithmic thinking, recursive thinking, and pattern recognition.

Moreover, Scratch visual programming language has been extensively used for teaching beginning programming for many different fields and ages. For instance, Scratch has been used to acquire programming skills in game-based problem solving in engineering education [28–30]. In addition, Scratch has been used teach programming to children in combination with certain metaphors [31]. However, these works and visual programming languages such as Scratch are still far from being useful for teaching advanced algorithmic techniques such as backtracking.

On the contrary, our proposed framework (i.e., implementing interactive online tools for showing examples of backtracking algorithms in which students can graphically observe execution step-by-step) tackles all these limitations and provides a collaborative platform for an experimental group of students to interact with the learning environment. Moreover, our solution provides increased learning when compared with the controlled group of students. In addition, engaging students to properly understand and program backtracking algorithms can contribute to the quality of future tools developed with this technique. To the best of our knowledge, no prior study covers the gap in the literature of visually teaching the advanced backtracking algorithmic with the framework proposed in the next section.

3. Framework for Teaching Algorithms with Interactive Applications

The framework is called Teaching Algorithms with Interactive Applications (TAI) and is implemented inside the file "tai.js". The framework is developed in JavaScript programming language, as most computers, laptops, and mobile devices have browsers supporting this language. In this way, students do not need to install anything in their computers, allowing them to be able to immediately use the educational applications developed with TAI.

The purpose of TAI is to support the development of interactive applications that allow students to observe and control the progress of algorithms, with special focus on backtracking algorithms. Therefore, TAI proposes to implement algorithms considering event-oriented progression of its executions triggered by user button-clicks. For this purpose, TAI stores functions by wrapping recursive calls in a stack. These recursive calls are executed sequentially by the student pressing a button. It uses a non-blocking mechanism by storing functions and associating button-pressing with execution and removal of the first recursive call in the queue.

The framework also stores in a stack the sets of parameters for each pending function call in the stack. In the design of recursive algorithms, all parameters are recommended to be included in one object with all the necessary fields. In this way, the algorithm designer can invoke the generic "wait" function with all the parameters in the same object. The "next" button is associated with execution of the next pending call with its parameters. As it is not exactly recursion, all array parameters need to be cloned if the algorithm uses multiple recursions with changing array values so the state of all parameters is properly stored.

For easy integration with other libraries and programs, TAI uses encapsulation by providing all its functionalities through an object called 'tai', generated by the Tai() function, which creates all the functions with necessary shared information.

TAI also provides functions for showing different types of parameters, such as arrays. For instance, the array of boolean values is shown with different colors for the two possible values and the written values, as one can observe in the N-queens case study.

Since TAI uses client programming, the algorithm is executed in the students' browsers and does not consume any server resources. This guarantees that many students can execute the program simultaneously without overloading the computing capability of the server. In addition, any web space can store the backtracking-algorithm interactive educational

applications without any particular application server, so TAI and its created applications could easily be widely used.

In order to facilitate the usage of the TAI framework by other professors and/or researchers, we explained the main aspects in a video, available online (TAI: JavaScript Framework for Teaching Algorithms interactively https://youtu.be/O9ZjVB-yzWQ (last accessed on 25 February 2022)).

4. Case Study with N-Queens Problem

We developed an interactive application with the proposed TAI framework for allowing students to understand the backtracking solution of the well-known n-queens problem [32], which is now used for solving other chess problems and is commonly used for teaching backtracking algorithms. In this problem, the goal is to place N queens on a chessboard of N × N cells so that there is not any pair of queens that can attack each other. In chess, queens can move through any number of cells in either a column, a row or a diagonal. Chessboards are usally composed of 8x8 cells, and this problem generalizes the 8-queen problem for any size of square chessboards.

Figure 1 shows the user interface of the N-queens system developed with the proposed framework. It illustrates the backtracking solution for the n-queens problem, showing an interactive step-by-step execution. The "size" text field allows students to set any size of chessboard, and after hitting the "start" button, the new chessboard is generated empty. Later on, the user can start observing the progress of the algorithm by hitting the "next" button to advance each step. In each step, if possible, a queen is placed. Otherwise, the algorithm goes backwards as much as necessary until it can place a queen in a new place, following the common execution scheme of backtracking algorithms.

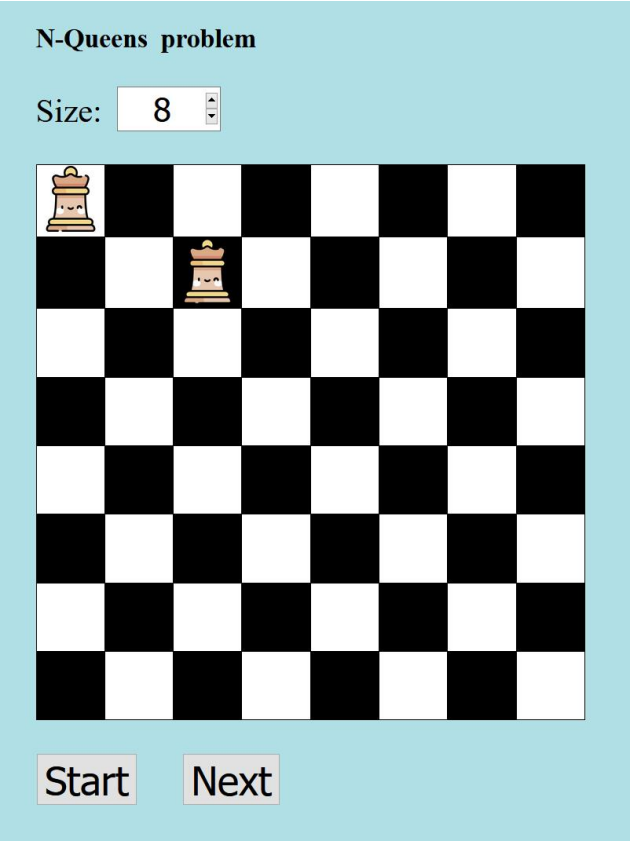


Figure 1. User interface of the n-queens interactive system.

For educational purposes, the application also shows graphically the key parameters of the backtracking solution so that students can understand their meaning and the evolution

of their values. Figure 2 shows an example of the n-queens interactive application reaching a solution. One can observe how the application shows all these parameters through the step-by-step execution.

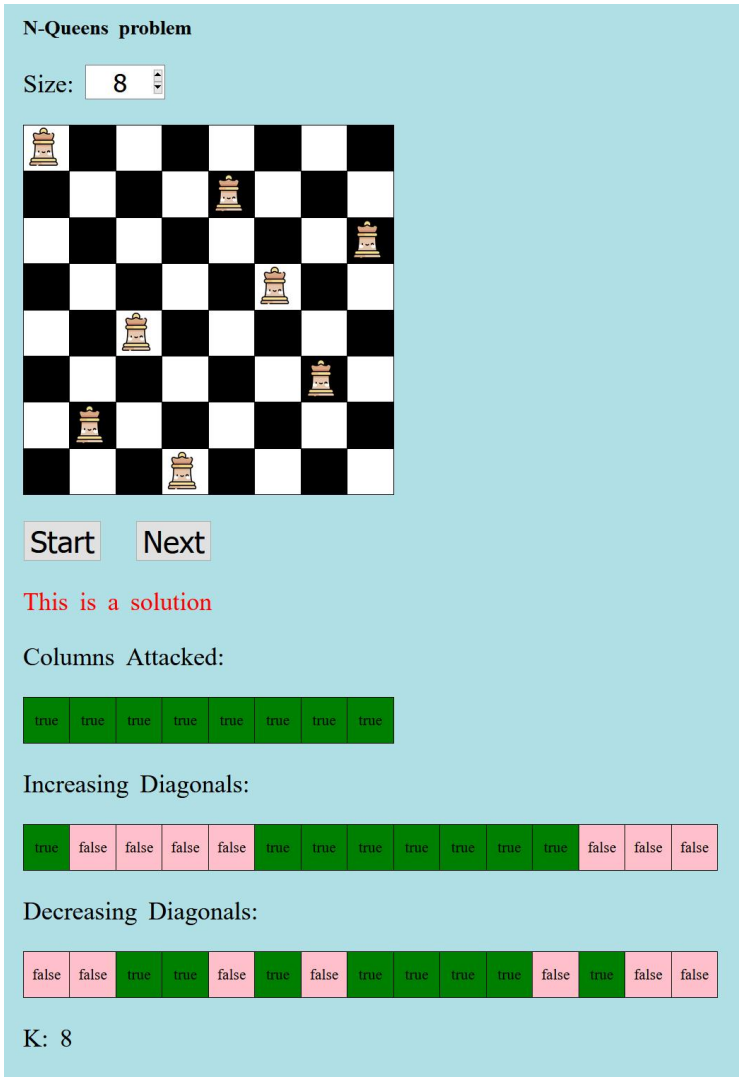


Figure 2. Example of reaching a solution in the n-queens interactive system.

This novel n-queens interactive application is available online (N-Queens interactive education tool http://grasia.fdi.ucm.es/ivan/nqueens/ (last accessed on 25 February 2022)), so other teachers, students, and researchers can benefit from it. Furthermore, we also created a video for introducing the N-queens problem to students (N-Queens: Introduction to the Problem https://youtu.be/Rxa_IJ3NbUM (last accessed on 25 February 2022)). Then, another set of our videos (N-Queens: Keys of the Backtracking Solution https://youtu.be/IKsAdWLPmhk (last accessed on 25 February 2022)) helps students understand key aspects of the backtracking solution for this problem with the proposed interactive application.

5. Experimentation

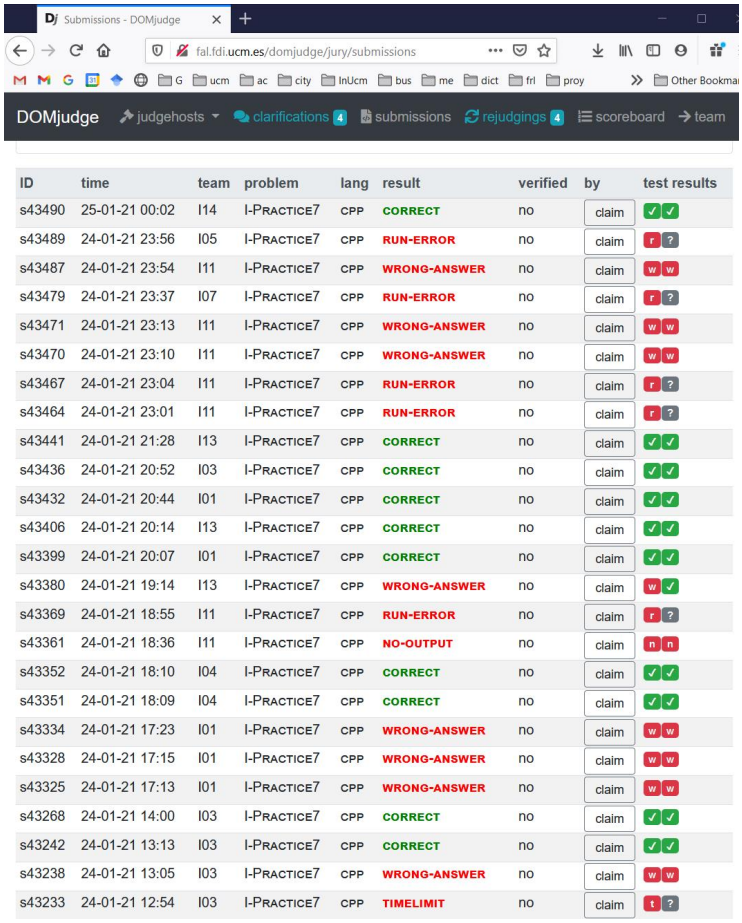
The n-queens interactive backtracking application was used with a group of 37 students from two different universities of two different countries: PSU in Saudi Arabia and UCM in Spain. They were 21.9 years old on average, with a standard deviation (SD) of 2.25. The students were studying different subjects related to programming, communications, and fundamentals of algorithms.

In this experiment, students were asked to watch the videos about (a) the introduction of the N-queens problem, (b) the keys to solving this problem with the tool, and (c) the principles of the TAI framework. The students were not forced to watch the videos completely, and they were encouraged to use the interactive online application.

Once they had learned using the interactive n-queens application developed with TAI, the students were surveyed with the System Usability Scale (SUS) [33] (the text of the SUS questionnaire is found in Appendix A) and the ease-of-learning and satisfaction dimensions of the Usefulness, Satisfaction and Ease-of-Use (USE) questionnaire [34] (the text of the USE questionnaire is found in Appendix B).

Backtracking learning was measured with a backtracking practice problem automatically corrected with software that tested not only visible input–output examples but also some hidden pairs of inputs and outputs. The results were compared with the results of a control group from the previous academic course that did not use the proposed framework but for whom every other teaching resource was the same. The backtracking practice problem was about finding the shortest path using backtracking in a matrix of cells in which a person could go only through the cells with odd numbers, since even numbers represented walls.

In order to illustrate the experience with more detail, Figure 3 shows the program we used to automatically correct the backtracking practice problems. In particular, we used the DOMjudge software, in which the students' algorithms were executed with some test input cases, and the output was compared with the corresponding cases. Since some of the test cases were hidden for students, they needed to solve the problem properly to pass the DOMjudge software. Performance of algorithms was automatically checked with a time limit for executing the test cases.



ID	time	team	problem	lang	result	verified	by	test results
s43490	25-01-21 00:02	I14	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43489	24-01-21 23:56	I05	I-Practice7	CPP	RUN-ERROR	no	claim	r ?
s43487	24-01-21 23:54	I11	I-Practice7	CPP	WRONG-ANSWER	no	claim	w w
s43479	24-01-21 23:37	I07	I-Practice7	CPP	RUN-ERROR	no	claim	r ?
s43471	24-01-21 23:13	I11	I-Practice7	CPP	WRONG-ANSWER	no	claim	w w
s43470	24-01-21 23:10	I11	I-Practice7	CPP	WRONG-ANSWER	no	claim	w w
s43467	24-01-21 23:04	I11	I-Practice7	CPP	RUN-ERROR	no	claim	r ?
s43464	24-01-21 23:01	I11	I-Practice7	CPP	RUN-ERROR	no	claim	r ?
s43441	24-01-21 21:28	I13	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43436	24-01-21 20:52	I03	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43432	24-01-21 20:44	I01	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43406	24-01-21 20:14	I13	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43399	24-01-21 20:07	I01	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43380	24-01-21 19:14	I13	I-Practice7	CPP	WRONG-ANSWER	no	claim	w ✓
s43369	24-01-21 18:55	I11	I-Practice7	CPP	RUN-ERROR	no	claim	r ?
s43361	24-01-21 18:36	I11	I-Practice7	CPP	NO-OUTPUT	no	claim	n n
s43352	24-01-21 18:10	I04	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43351	24-01-21 18:09	I04	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43334	24-01-21 17:23	I01	I-Practice7	CPP	WRONG-ANSWER	no	claim	w w
s43328	24-01-21 17:15	I01	I-Practice7	CPP	WRONG-ANSWER	no	claim	w w
s43325	24-01-21 17:13	I01	I-Practice7	CPP	WRONG-ANSWER	no	claim	w w
s43268	24-01-21 14:00	I03	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43242	24-01-21 13:13	I03	I-Practice7	CPP	CORRECT	no	claim	✓ ✓
s43238	24-01-21 13:05	I03	I-Practice7	CPP	WRONG-ANSWER	no	claim	w w
s43233	24-01-21 12:54	I03	I-Practice7	CPP	TIMELIMIT	no	claim	t ?

Figure 3. DOMjudge used for automatically correcting backtracking practice problems.

Figure 4 compares the learning of the experimental group that used the proposed TAI approach to the control group. Learning was measured as the percentage of teams that passed the backtracking practice problem according to the automatic judge. The experimental group scored 66.67%, while the control group obtained 60.00%. Thus, the proposed approach increased performance by 6.67%.

According to the YouTube statistics, on average, the students watched only 24.4% of the first video, 5.9% of the second video, and 8.4% of the third video. This may reveal that students engaged soon with the application once they understood the problem, without needing to see the entirety of the videos, showing that they probably found it easy to start using the application.

The results of SUS were 74.5% on average, with a SD of 13.3%. The results of USE were 6.06 out of 7 on average, with a SD of 1.11. In particular, USE obtained 6.22 out of 7 on average for ease-of-learning (4 questions), and 5.97 out of 7 on average for satisfaction (7 questions). Thus, the highest-evaluated aspect of the application was its ease-of-learning.

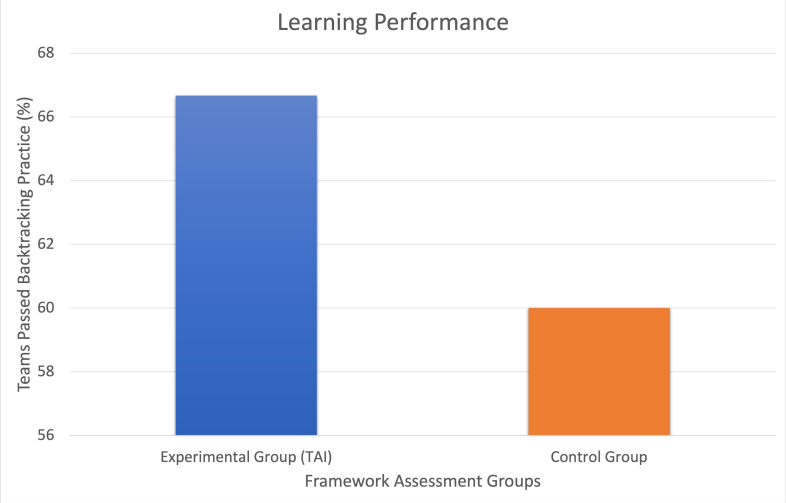


Figure 4. Comparison of learning performance based on automatic correction of algorithms in a backtracking practice problem between experimental group (TAI) and control group.

Individual analysis of some items of the USE scale revealed some aspects of the perception of the students when using the n-queens application. More specifically, Figure 5 shows the seven-point Likert responses about whether students learned to use the interactive tool quickly.

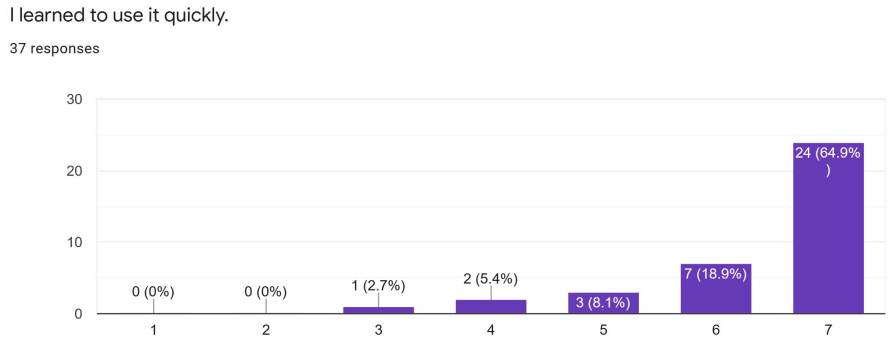


Figure 5. Seven-point Likert responses to "I learned to use it quickly".

Considering the possibility of peer-to-peer recommendation among students, Figure 6 presents the seven-point Likert replies to whether the students would recommend it to a friend.

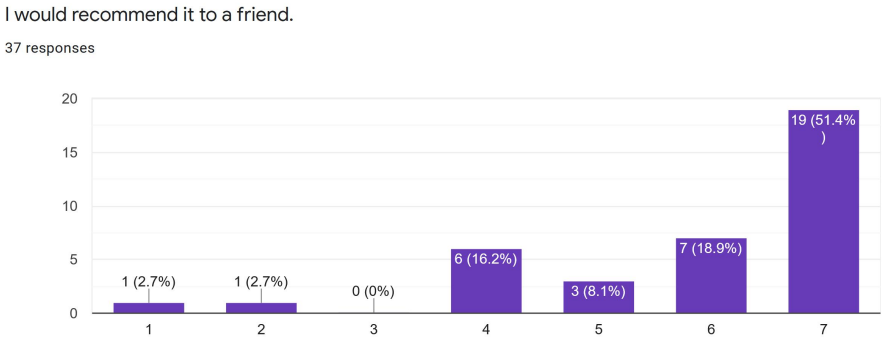


Figure 6. Seven-point Likert responses to "I would recommend it to a friend".

Regarding the entertainment aspect, Figure 7 shows students enjoyed using this interactive tool for learning, with 69.6% in complete agreement (i.e., 7 out of 7) and 25.7% in agreement (i.e., 6 out of 7).

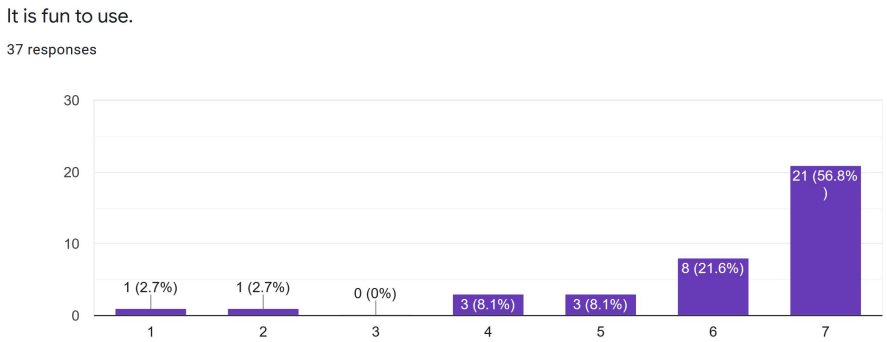


Figure 7. Seven-point Likert responses to "It is fun to use".

Taking student expectations into account, Figure 8 confirms that the interactive tool mostly met students expectations, as they mainly agree that the tool worked as they wanted it to work.

Furthermore, Figure 9 shows that most students agreed that they felt they needed this kind of tool for learning these kinds of algorithms.

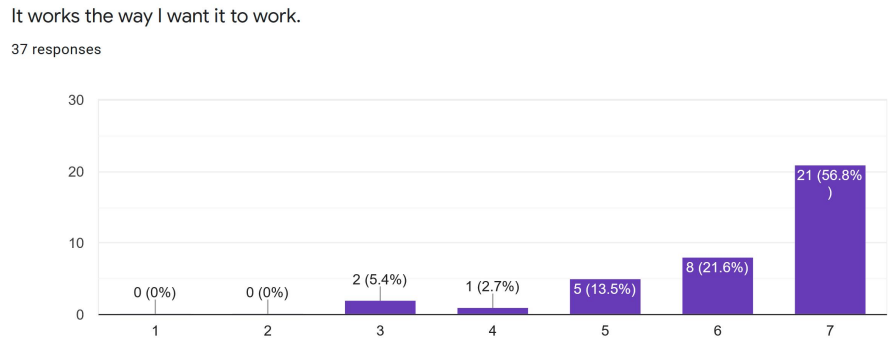


Figure 8. Seven-point Likert responses to "It works the way I wanted it to work".

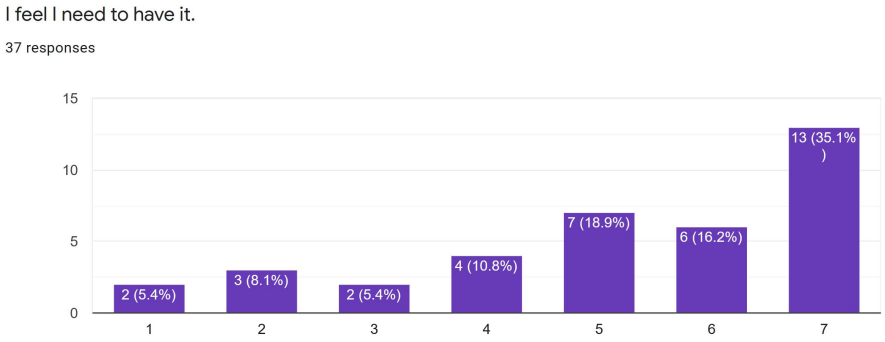


Figure 9. Seven-point Likert responses to "I feel I need to have it".

Finally, Table 1 presents the average results out of seven for each question of USE to provide a higher level overview of all the replies, considering each individual question of the USE scale.

Table 1. USE average results for each question in seven-point Likert scale .

Index	USE item	Score out of 7
1	I learned to use it quickly.	6.36
2	I easily remember how to use it.	6.08
3	It is easy to learn to use it.	6.39
4	I quickly became skillful with it.	6.06
5	I am satisfied with it.	6.17
6	I would recommend it to a friend.	5.86
7	It is fun to use.	6.11
8	It works the way I want it to work.	6.25
9	It is wonderful.	6.03
10	I feel I need to have it.	5.31
11	It is pleasant to use.	6.08

6. Discussion

The TAI framework has shown its utility for developing interactive educative applications for teaching backtracking algorithms. The proposed n-queen interactive application case study has shown its impact on students' backtracking learning, which increased by 6.67% in the experimental group over the control group. Experimentation also showed general acceptance among students, who showed their satisfaction and perception of ease of learning through the validated USE scale.

A possible limitation of the novel TAI framework is that teachers may need some time to adapt their backtracking examples to be taught interactively, as the proper usage of TAI in new applications may require (a) programming the visualization specific to a given problem and (b) some basic knowledge of JavaScript for using the framework. Innovation in education usually involves effort from teachers and innovators, as can be observed in different fields such as the inclusion of 3D technology in healthcare education [18], in which 3D models needed to be developed to be either visualized or printed. In the field of computer science, other frameworks about visualization of hardware simulation, such as the one presented by [20], required expressing hardware organization in a specific modeling language. This barrier to our presented TAI framework can be ameliorated by creating a repository of backtracking examples so that effort is shared among teachers. This will facilitate reusing some components for different problems.

In different user experience (UX) evaluation frameworks, such as [21], speed to learn plays a crucial role in the time for doing some tasks for the first time in teaching–learning processes, and usually this time influences the perception of the learning experience. The average result of 6.57 out of 7 for the item "I learned to use it quickly" reveals that the proposed TAI framework allows one to create interactive tools that are really fast to learn, aligning with common principles for a great student learning experience.

Recommendations by users to their friends is one of the key influencing factors on the popularity of web applications [35]. The high average result of 6.22 out of 7 for the item "I would recommend it to a friend" reveals the potential of TAI for developing popular web applications for teaching and learning algorithms.

The systematic mapping of gamification in software engineering in [19] revealed that playful and fun learning experiences can both increase student motivation and, in some cases, even their learning performance. However, this review also highlighted that the literature still lacks proper playful and fun applications for teaching and learning certain aspects of software development. In this context, our work presents a case study of an interactive tool for learning an algorithm, which was ranked considerably high on average for the item regarding fun. In particular, this work has illustrated this with backtracking, which is one of the algorithm techniques that is usually perceived as most difficult and complex.

The gap between student expectations and reality in software engineering education can become a drawback, as [36] stated. The application met the expectations of students, as revealed by the high average score for the item "It works the way I wanted it to work", overcoming this usual barrier in the field of education in software development.

Engagement in learning tasks is key for continuous and fruitful learning experiences [37]. The high average score for the item "I feel I need to have it" showed that students were engaged from the beginning with the interactive application developed with the proposed TAI framework.

Since the experiments in PSU were based on watching videos and using the interactive application without any detailed face-to-face introduction of backtracking beforehand, the proposed TAI framework will probably be useful for creating material for flipped classrooms. This aligns with the current trend of flipped classrooms in engineering education shown in works such as [9], which has been proven to increase student responsibility and learning outcomes in a language processors course. In flipped classrooms, students study before lectures with some motivating material so that lectures can be more advanced and dedicated to more complex exercises and answering relevant questions. In addition, educational software tools have proven to be useful in different engineering fields, such as the Pardos tool [22], which was successfully used for teaching sound propagation analysis to engineering students. The proposed interactive application developed with TAI also aligns with this research for developing educational software.

7. Conclusions

The experiments showed that TAI framework supported the development of an interactive tool for teaching the backtracking algorithm for solving the n-queens problem. Backtracking learning was increased in the experimental group over the control group, as shown in the results of a backtracking practice problem which was automatically corrected by a judge. According to the USE scale, the system was easy to learn, and students felt satisfied with their learning experience. For some individual items, we also observed students enjoyed the learning experience, and the application met student expectations. In fact, they felt that this kind of interactive tool was necessary to learn such types of algorithms, and they expressed that they would recommend the application to a friend. The proposed framework advocates to be cross-cultural, as it was successfully applied to two countries with different cultures.

Our most relevant future work is to promote the usage of TAI with a public repository in order to foster a community around this framework for improving teaching and learning

of difficult algorithms. As an example, we plan to develop another case study about the Dijkstra algorithm for finding the shortest path, supporting visualization of graphs, and making it easier to develop interactive tools for teaching algorithms in graphs.

Funding: This research received no external funding.

Acknowledgments: The author would like to acknowledge Prince Sultan University and Smart Systems Engineering Lab for their valuable support. Further, the authors would like to acknowledge the support of Prince Sultan University for article processing charges (APC) for this publication.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A

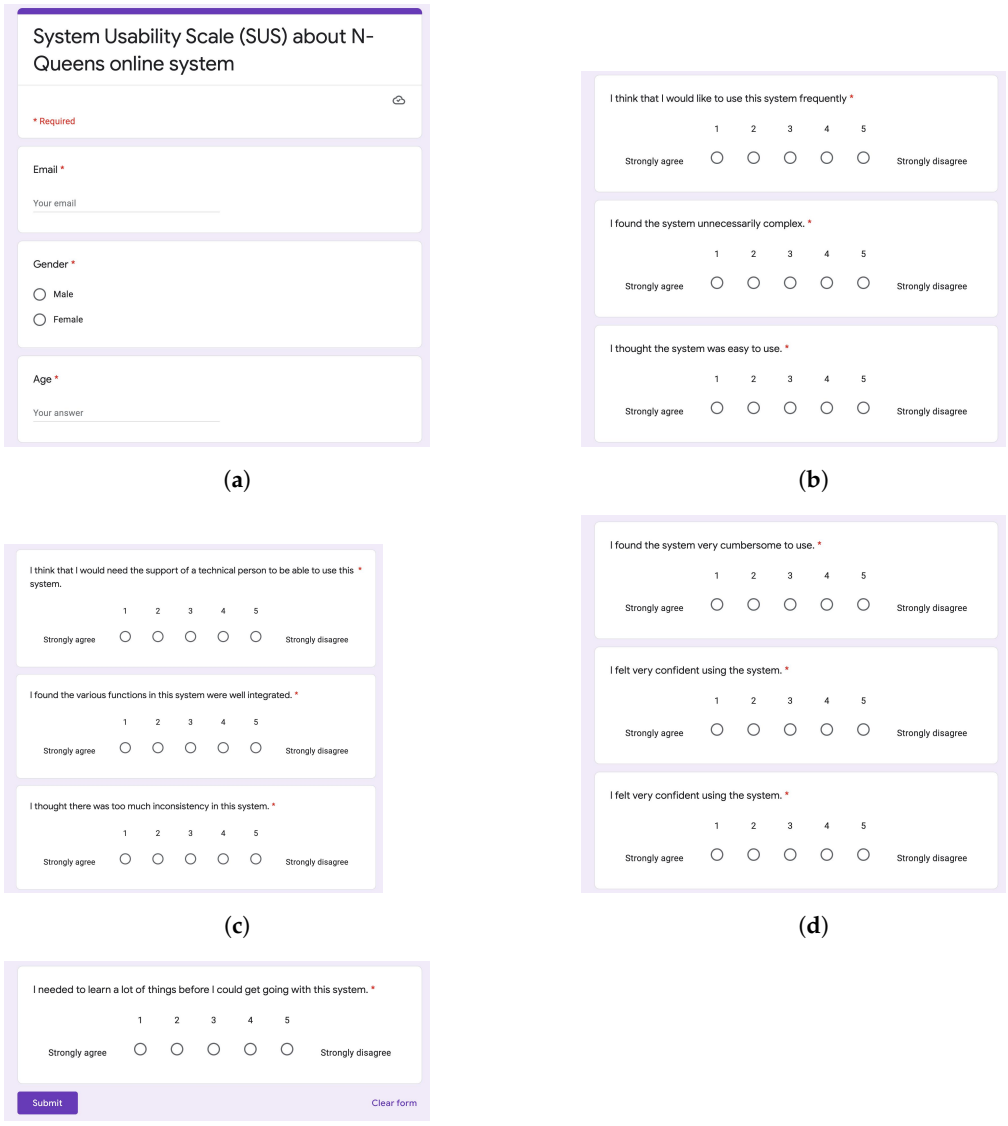
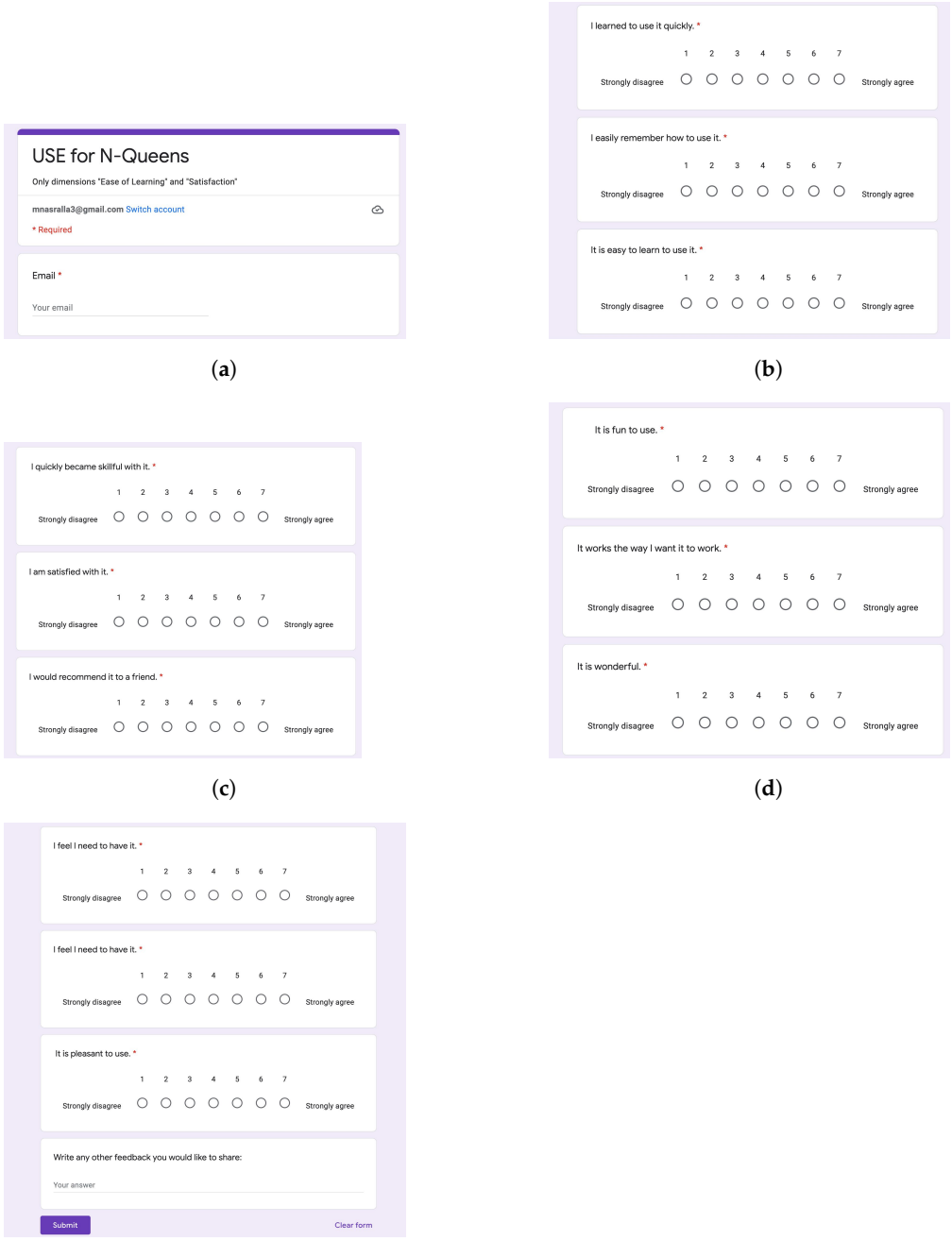


Figure A1. System Usability Scale (SUS) for n-queens online system questionnaire. (**a**) SUS questionnaire page 1; (**b**) SUS questinaire page 2; (**c**) SUS questinaire page 3; (**d**) SUS questinaire page 4; (**e**) SUS questinaire page 5.

Appendix B



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure A2. Usefulness, Satisfaction and Ease-of-Use (USE) for n-queens (*we only used the "Ease of Learning" and "Satisfaction" dimensions*) questionnaire. (**a**) USE questionnaire page 1; (**b**) USE questionnaire page 2; (**c**) USE questionnaire page 3; (**d**) USE questionnaire page 4; (**e**) USE questionnaire page 5.

References

1. Höddinghaus, M.; Sondern, D.; Hertel, G. The Automation of Leadership Functions: Would People Trust Decision Algorithms? *Comput. Hum. Behav.* **2021**, *116*, 106635. [CrossRef]
2. Lu, X.; Wang, X.; Hui, P. DADIM: A distance adjustment dynamic influence map model. *Future Gener. Comput. Syst.* **2020**, *112*, 1122–1130. [CrossRef]
3. Wedde, H.F.; Farooq, M. A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks. *J. Syst. Archit.* **2006**, *52*, 461–484. [CrossRef]

4. Liu, Y.A.; Stoller, S.D. From recursion to iteration: What are the optimizations? In Proceedings of the 2000 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation, Boston, MA, USA, 22–23 January 1999; pp. 73–82.
5. Gu, M.; Eisenstat, S.C. A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. *SIAM J. Matrix Anal. Appl.* **1995**, *16*, 172–191. [CrossRef]
6. Priestley, H.A.; Ward, M.P. A multipurpose backtracking algorithm. *J. Symb. Comput.* **1994**, *18*, 1–40. [CrossRef]
7. Haq, I.U.; Anwar, A.; Rehman, I.U.; Asif, W.; Sobnath, D.; Sherazi, H.H.R.; Nasralla, M.M. Dynamic Group Formation With Intelligent Tutor Collaborative Learning: A Novel Approach for Next Generation Collaboration. *IEEE Access* **2021**, *9*, 143406–143422. [CrossRef]
8. Singh, N.; Gunjan, V.K.; Nasralla, M.M. A parametrized comparative analysis of performance between proposed adaptive and personalized tutoring system "seis tutor" with existing online tutoring system. *IEEE Access* **2022**, *10*, 39376–39386. [CrossRef]
9. Urquiza-Fuentes, J. Increasing students' responsibility and learning outcomes using partial Flipped Classroom in a Language Processors course. *IEEE Access* **2020**, *8*, 211211–211223. [CrossRef]
10. Li, Z.; Li, T.; Wu, Y.; Yang, L.; Miao, H.; Wang, D. Software Defect Prediction Based on Hybrid Swarm Intelligence and Deep Learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 4997459. [CrossRef]
11. Statter, D.; Armoni, M. Teaching abstraction in computer science to 7th grade students. *ACM Trans. Comput. Educ. (TOCE)* **2020**, *20*, 1–37. [CrossRef]
12. Boyd, T.M.; Romig, P.R. Cross-disciplinary education: The use of interactive case studies to teach geophysical exploration. *Comput. Geosci.* **1997**, *23*, 593–599. [CrossRef]
13. Nasralla, M.M.; Al-Shattarat, B.; Almakhles, D.J.; Abdelhadi, A.; Abowardah, E.S. Futuristic Trends and Innovations for Examining the Performance of Course Learning Outcomes Using the Rasch Analytical Model. *Electronics* **2021**, *10*, 727. [CrossRef]
14. Kiss, G.; Arki, Z. The influence of game-based programming education on the algorithmic thinking. *Procedia-Soc. Behav. Sci.* **2017**, *237*, 613–617. [CrossRef]
15. Zhang, Y.; Huang, C.; Jin, Z. Backtracking search algorithm with reusing differential vectors for parameter identification of photovoltaic models. *Energy Convers. Manag.* **2020**, *223*, 113266. [CrossRef]
16. Yogesh, P.R. Backtracking Tool Root-Tracker to Identify True Source of Cyber Crime. *Procedia Comput. Sci.* **2020**, *171*, 1120–1128. [CrossRef]
17. Calderón, A.; Ruiz, M.; O'Connor, R.V. A multivocal literature review on serious games for software process standards education. *Comput. Stand. Interfaces* **2018**, *57*, 36–48. [CrossRef]
18. Alhonkoski, M.; Salminen, L.; Pakarinen, A.; Veermans, M. 3D technology to support teaching and learning in health care education–A scoping review. *Int. J. Educ. Res.* **2021**, *105*, 101699. [CrossRef]
19. Alhammad, M.M.; Moreno, A.M. Gamification in software engineering education: A systematic mapping. *J. Syst. Softw.* **2018**, *141*, 131–150. [CrossRef]
20. Garay, G.R.; Tchernykh, A.; Drozdov, A.Y.; Garichev, S.N.; Nesmachnow, S.; Torres-Martinez, M. Visualization of VHDL-based simulations as a pedagogical tool for supporting computer science education. *J. Comput. Sci.* **2019**, *36*, 100652. [CrossRef]
21. Alomari, H.W.; Ramasamy, V.; Kiper, J.D.; Potvin, G. A User Interface (UI) and User eXperience (UX) evaluation framework for cyberlearning environments in computer science and software engineering education. *Heliyon* **2020**, *6*, e03917. [CrossRef]
22. Pardo-Quiles, D.; Rodríguez, J.V.; Rodríguez-Rodríguez, I. PARDOS: An Educational Software Tool for the Analysis of Sound Propagation. *IEEE Access* **2020**, *8*, 194933–194949. [CrossRef]
23. Baerten, H.; Van Reeth, F. Using VRML and JAVA to visualize 3D algorithms in computer graphics education. *Comput. Netw. ISDN Syst.* **1998**, *30*, 1833–1839. [CrossRef]
24. Agbo, F.J.; Oyelere, S.S.; Suhonen, J.; Tukiainen, M. iThinkSmart: Immersive Virtual Reality Mini Games to Facilitate Students' Computational Thinking Skills. In Proceedings of the Koli Calling'21: 21st Koli Calling International Conference on Computing Education Research, Joensuu, Finland, 18–21 November 2021; Association for Computing Machinery: New York, NY, USA, 2021.
25. Toivonen, T.; Oyelere, S. BiGO: A Toolset to Support CS Students to Learn to Analyze Time Complexities of Algorithms. In Proceedings of the 19th Koli Calling International Conference on Computing Education Research, Koli, Finland, 21–24 November 2019; pp. 1–6.
26. Agbo, F.J. Co-Designing a Smart Learning Environment to Facilitate Computational Thinking Education in the Nigerian Context. Ph.D. Thesis, Itä-Suomen yliopisto, Kuopio, Finland, 2022.
27. Sobnath, D.; Rehman, I.U.; Nasralla, M.M. Smart cities to improve mobility and quality of life of the visually impaired. In *Technological Trends in Improved Mobility of the Visually Impaired*; Springer: Cham, Switzerland, 2020; pp. 3–28.
28. Oshanova, N.; Anuarbekova, G.; Shekerbekova, S.; Arynova, G. Algorithmization and Programming Teaching Methodology in the course of Computer Science of Secondary School. *Aust. Educ. Comput.* **2019**, *34*, 1–14.
29. Topalli, D.; Cagiltay, N.E. Improving programming skills in engineering education through problem-based game projects with Scratch. *Comput. Educ.* **2018**, *120*, 64–74. [CrossRef]
30. Rehman, I.U.; Sobnath, D.; Nasralla, M.M.; Winnett, M.; Anwar, A.; Asif, W.; Sherazi, H.H.R. Features of mobile apps for people with autism in a post COVID-19 scenario: Current status and recommendations for apps using AI. *Diagnostics* **2021**, *11*, 1923. [CrossRef]
31. Pérez-Marín, D.; Hijón-Neira, R.; Bacelo, A.; Pizarro, C. Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Comput. Hum. Behav.* **2020**, *105*, 105849. [CrossRef]

32. Kolossoski, O.; Matioli, L.; Torrealba, E.; Silva, J. Modular knight distance in graphs and applications on the n-queens problem. *Discret. Math.* **2020**, *343*, 112136. [CrossRef]
33. Peres, S.C.; Pham, T.; Phillips, R. Validation of the system usability scale (SUS) SUS in the wild. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, San Diego, CA, USA, 30 September–4 October 2013; SAGE Publications Sage CA: Los Angeles, CA, USA, 2013; Volume 57, pp. 192–196.
34. Gao, M.; Kortum, P.; Oswald, F. Psychometric evaluation of the use (usefulness, satisfaction, and ease of use) questionnaire for reliability and validity. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Philadelphia, PA, USA, 1–5 October 2018; SAGE Publications Sage CA: Los Angeles, CA, USA, 2018; Volume 62, pp. 1414–1418.
35. Moniz, N.; Torgo, L. A review on web content popularity prediction: Issues and open challenges. *Online Soc. Netw. Media* **2019**, *12*, 1–20. [CrossRef]
36. Iacob, C.; Faily, S. Exploring the gap between the student expectations and the reality of teamwork in undergraduate software engineering group projects. *J. Syst. Softw.* **2019**, *157*, 110393. [CrossRef]
37. Clynes, M.; Sheridan, A.; Frazer, K. Student engagement in higher education: A cross-sectional study of nursing students' particpation in college-based education in the republic of Ireland. *Nurse Educ. Today* **2020**, *93*, 104529. [CrossRef] [PubMed]