

Cogniquist - DSA Gamified Learning Platform

Manimegala M ^{#1}, Pooja P^{#2}, Parinita Roshan V^{#3}, Prahathi S^{#4}, Reshma K^{#5}, Ranjana S^{#6}

^{#1}Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

^{#2}Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

^{#3}Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

^{#4}Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

^{#5}Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

^{#6}Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

ABSTRACT

COGNIQUEST – DSA Game Based Learning App is an interactive educational application designed to make learning Data Structures and Algorithms (DSA) more engaging, practical and enjoyable for students. Data Structures and Algorithms are fundamental concepts in computer science and play an important role in programming, problem-solving, technical interviews and placement preparation. However, many students find DSA difficult to understand because traditional learning methods mainly focus on theoretical explanations and repetitive coding exercises. COGNIQUEST addresses this challenge by introducing a game-based learning approach that combines DSA concepts with interactive challenges, levels, rewards, progress tracking and achievement-based learning.

I. INTRODUCTION

In the modern digital era, technology has become an important part of education, particularly in the field of computer science. The growth of software development, artificial intelligence, data science, cybersecurity and other technology-based industries has increased the demand for students with strong programming and problem-solving skills. Among the fundamental concepts of computer science, Data Structures and Algorithms (DSA) play a significant role in developing efficient programs and solving computational problems.

A strong understanding of DSA is essential for students who wish to build a career in software development and related technical fields. It is also an important area of preparation for coding assessments, technical interviews, competitive programming and placement examinations. However, learning DSA can be challenging for many students because the concepts involve logical thinking, algorithmic reasoning, mathematical understanding and continuous programming practice.

Data Structures are methods used to organize and store data so that it can be accessed and processed efficiently, while Algorithms are step-by-step procedures used to solve specific problems. Concepts such as arrays, strings, linked lists, stacks, queues, searching, sorting, recursion, trees, graphs and other algorithmic techniques form the foundation of programming and software engineering. Students are generally introduced to these concepts through classroom lectures, textbooks, online tutorials, videos and coding practice platforms. Although these resources provide valuable knowledge, many learners still experience difficulties in understanding abstract concepts and applying them to practical problems.

Learning DSA only through theoretical explanations can sometimes make the subject less interesting, especially for beginners. Students may understand the definition of a data structure or algorithm but struggle when they are asked to implement it or select an appropriate approach for solving a problem.

One of the major challenges in DSA education is maintaining student motivation and consistency. Learning algorithms requires repeated practice and solving programming problems often involves trial and error. When students encounter difficult questions repeatedly, they may lose confidence or interest in continuing their practice.

Traditional learning environments generally focus on completing lessons, assignments and examinations, whereas students may benefit from a more interactive environment that provides continuous encouragement and immediate feedback. Therefore, there is a need for an educational platform that can make DSA learning more engaging while still maintaining its technical and academic purpose.

Game-based learning provides an effective approach for addressing this challenge. It combines educational content with elements commonly found in games .

Overall, COGNIQUEST – DSA Game Based Learning App represents an innovative approach to learning Data Structures and Algorithms. It provides a structured learning journey supported by interactive challenges, missions, levels, XP, badges, lives, streaks, rewards, progress tracking and leaderboards. The system aims to improve student engagement, encourage consistent practice, strengthen logical thinking and support the development of problem-solving abilities.

II. LITERATURE SURVEY

Data Structures and Algorithms (DSA) are fundamental components of computer science education. They help students understand how data can be organized, processed and manipulated efficiently and how computational problems can be solved using systematic procedures. Concepts such as arrays, linked lists, stacks, queues, trees, graphs, searching, sorting and recursion form an essential foundation for programming and software development. As DSA is also commonly included in technical interviews and coding assessments, students are required to develop both conceptual knowledge and practical problem-solving skills.

Traditional DSA education generally depends on classroom teaching, textbooks, lecture notes, online tutorials and programming exercises. These approaches are useful for introducing theoretical concepts, but students may experience difficulty when attempting to apply those concepts to programming problems. Therefore, researchers and educators have explored interactive learning methods that can improve student understanding, participation and motivation.

Game-based learning is one such approach that has gained attention in educational technology. It uses game mechanics and interactive activities to support educational objectives. Applying this approach to DSA can make learning more engaging by allowing students to learn concepts through challenges, rewards, levels and continuous feedback.

Game-Based Learning in Education : Game-based learning refers to the use of games or game-like environments for achieving educational goals. Unlike traditional teaching methods, game-based learning allows learners to actively participate in activities instead of simply receiving information. Games can provide challenges, immediate feedback, rewards and clear goals which can encourage learners to remain involved in the learning process.

Gamification in Learning Applications: Gamification is the process of incorporating game-related elements into non-game environments to increase engagement and motivation. In educational applications, common gamification elements

include points, levels, badges, rewards, challenges, progress bars, streaks and leaderboards.

Interactive Learning for Data Structures: Interactive learning provides students with opportunities to actively participate in the learning process. In DSA education, interactive visualizations can help learners understand how data structures operate internally. For example, a visualization of a stack can demonstrate push and pop operations, while a linked-list visualization can show how nodes are connected and modified.

Such visual and interactive representations can make abstract concepts easier to understand. Instead of only reading an algorithm, learners can observe how data changes when an operation is performed. This can improve conceptual understanding and help students connect theoretical knowledge with practical implementation.

Learning Through Challenges and Quizzes: Challenges and quizzes are commonly used in digital learning systems to evaluate students' understanding. They provide an opportunity for learners to apply the concepts they have studied and identify areas where they require additional practice.

COGNIQUEST uses a mission-based approach where learners can complete challenges as part of their learning journey. Completing challenges can contribute to XP and progression. This provides learners with an immediate objective and a visible reward for their effort.

Progress Tracking and Performance Monitoring: Progress tracking is an important feature of modern educational applications. It allows learners to understand their current position and identify completed and incomplete learning activities. Progress can be represented through percentages, levels, XP, completed missions, achievements and other indicators.

Without progress tracking, learners may find it difficult to understand how much of a course they have completed. A visible progress system can provide motivation because learners can observe their improvement over time.

Research Gap

The literature and existing learning approaches indicate that DSA can be taught through various digital methods, including online tutorials, coding platforms, visualizations, quizzes and interactive learning systems.

Leaderboard-Based Learning: Leaderboards are commonly used in gamified applications to display the relative performance of users. In educational systems, they can

encourage healthy competition and motivate students to improve their performance.

The literature survey shows that traditional DSA learning methods can be improved through interactive and engaging approaches. Game-based learning provides opportunities to increase participation, while gamification introduces motivational elements such as points, levels, badges, rewards, streaks and leaderboards.

III. PROPOSED METHODOLOGY

The proposed system, COGNIQUEST – DSA Game-Based Learning App, is designed to improve the learning experience of Data Structures and Algorithms (DSA) through an interactive and game-based educational approach. Data Structures and Algorithms are fundamental components of computer science education and are essential for programming, logical reasoning, algorithmic problem-solving, and technical interview preparation. However, many students find these concepts challenging because they require continuous practice, analytical thinking, and the ability to apply theoretical knowledge to practical problems. Traditional learning methods often depend on lectures, textbooks, and conventional coding exercises, which may not always maintain student motivation and consistent participation. To address these challenges, COGNIQUEST integrates educational content with gamification techniques to create a structured, engaging, and learner-centered environment.

The primary objective of the proposed methodology is to make DSA learning more accessible, motivating, and practice-oriented. The system organizes learning activities into structured topics, missions, challenges, and progressive levels. Students can study fundamental concepts, complete topic-based activities, earn experience points (XP), unlock achievements, and monitor their learning progress. Game elements such as rewards, lives, daily streaks, badges, and leaderboards are incorporated to encourage regular participation and continuous improvement. Through this approach, the application aims to connect theoretical learning with practical problem-solving while providing students with a clear and motivating learning pathway.

A. System Architecture

The proposed system follows a modular application architecture consisting of the user interface module, application logic module, database module, learning content module, gamification module, and progress tracking module. These components work together to provide an integrated learning experience.

The user interface module enables students to access learning materials, navigate DSA topics, participate in missions, answer questions, and view their achievements. The application logic module manages the rules associated with learning activities, level progression, scoring, rewards, and user interactions. The database module stores relevant information, such as user profiles, learning content, completed missions, XP, badges, and progress records, according to the application's implementation. The learning content module organizes DSA concepts into manageable learning units, while the gamification module manages the game-related elements that encourage student participation. The progress tracking module presents learners with information about their completed activities and achievements.

The modular architecture allows the different components to function together while supporting future enhancements. Additional features, such as interactive algorithm visualization, AI-based personalized recommendations, and automated coding evaluation, could be incorporated in future versions.

B. User Registration and Profile Management

The learning process begins when a student accesses the application and enters the learning environment. User registration and authentication can be used to provide individual access to the platform and maintain separate learning records for each student. After accessing the system, learners can view their profiles and dashboards, which provide information about their current learning status and achievements.

Profile management helps associate learning activities with individual users. Information such as completed topics, earned XP, unlocked badges, achieved levels, and mission completion status can be maintained to support progress monitoring. This allows learners to continue their learning journey and review their previous achievements. Individual progress records also provide a foundation for introducing more advanced learning analytics in future versions.

C. Structured DSA Learning Modules

The proposed application organizes Data Structures and Algorithms into structured learning modules. These modules cover important topics such as arrays, strings, linked lists, stacks, queues, searching algorithms, sorting algorithms, recursion, trees, and graphs. The learning content is arranged to support gradual progression from fundamental concepts to more complex topics.

Each module introduces the relevant concepts and provides opportunities for learners to reinforce their understanding through practice activities. Breaking the subject into smaller

learning units helps students approach complex topics systematically rather than attempting to study all concepts simultaneously. Learners can complete activities associated with a topic and progress through the learning pathway according to the application's progression rules.

This structured organization supports systematic revision and encourages students to build a strong foundation before attempting advanced problem-solving activities. It also provides a framework for extending the application with additional DSA topics in future development.

D. Game-Based Learning and Mission Design

A major component of COGNIQUEST is its mission-based learning approach. Instead of presenting learning activities only as conventional exercises, the application organizes selected tasks into missions that learners can complete as part of their learning journey. Each mission can focus on a particular concept, skill, or learning objective.

Missions may include concept-based questions, quizzes, problem-solving challenges, and topic-specific tasks, depending on the available implementation. Completing these activities enables learners to demonstrate their understanding and make progress through the learning system. Different missions can be associated with different levels of difficulty to support learners with varying levels of DSA knowledge.

The mission-based structure provides students with clear learning goals and encourages them to complete activities consistently. By combining educational tasks with game-related progression, the application seeks to make regular DSA practice more engaging and manageable.

E. Gamification Mechanism

The gamification module introduces game elements into the learning process to encourage student motivation and participation. COGNIQUEST incorporates features such as experience points (XP), levels, achievement badges, rewards, lives, daily streaks, and leaderboards.

Experience points are awarded according to the application's scoring rules when learners complete eligible activities. Accumulated XP can contribute to level progression, providing a visible representation of learning activity and achievement. Achievement badges recognize specific milestones, while rewards provide additional recognition for completing selected tasks or reaching predefined goals.

The lives mechanism introduces a game-like challenge into supported activities, while daily streaks encourage students to maintain consistent learning habits. Leaderboards display comparative performance among participating learners and can encourage friendly competition. Together, these elements

aim to create a more engaging learning environment while maintaining the educational focus of the application.

The gamification mechanism is intended to complement the learning content rather than replace conceptual understanding or practical programming experience. The overall objective is to encourage students to participate actively, practice regularly, and work toward clearly defined learning goals.

IV. SYSTEM IMPLEMENTATION

The COGNIQUEST – DSA Game-Based Learning App is implemented as an interactive educational platform designed to make learning Data Structures and Algorithms more engaging. The application provides a user-friendly interface through which students can access learning modules, complete missions, participate in challenges, and monitor their achievements. The system organizes DSA topics into structured learning activities to support step-by-step learning and regular practice.

The application incorporates gamification features such as experience points (XP), levels, achievement badges, rewards, lives, and daily streaks. These features encourage students to participate actively and maintain consistency in their learning activities. Students can complete missions and challenges to earn rewards and progress through different levels. The leaderboard also supports healthy competition by displaying learners' achievements according to the implemented ranking mechanism.

The progress tracking module maintains information about completed missions, earned XP, unlocked badges, achieved levels, and learning activities. This information enables students to monitor their performance and identify topics that require additional practice. The application integrates its learning modules and gamification features to provide a structured learning experience. The stored information is managed according to the application's implemented data management mechanism.

V. ADVANTAGES

1. Interactive Learning Experience

The application provides an interactive environment that makes learning Data Structures and Algorithms more interesting through game-based activities and challenges.

2. Improved Student Engagement

Game elements such as XP, badges, rewards, missions, and levels encourage students to participate actively and maintain their interest in learning.

3. Step-by-Step Learning

The application organizes DSA concepts into structured

learning levels, enabling students to understand fundamental concepts before progressing to advanced topics.

4. Enhanced Problem-Solving Skills

Regular practice through missions and challenges helps students develop logical thinking, algorithmic reasoning, and programming skills.

5. Motivation Through Gamification

Features such as achievement badges, experience points, lives, and daily streaks motivate students to continue learning and complete their learning goals.

6. Progress Tracking

The application enables students to monitor completed topics, earned XP, achievements, and learning progress, helping them identify areas that require further improvement.

7. Support for Placement Preparation

Practicing DSA concepts regularly helps students strengthen their programming knowledge and prepare for coding assessments, technical interviews, and placement examinations.

8. Self-Paced Learning

Students can learn and practice concepts at their own pace, allowing them to revisit difficult topics and improve their understanding through repeated practice.

9. Healthy Competition

Leaderboards encourage friendly competition among learners and motivate them to improve their performance while completing learning challenges.

10. Bridging Theory and Practice

The application connects theoretical DSA concepts with practical problem-solving activities, helping students apply their knowledge to programming-related problems.

VI. RESULTS AND ANALYSIS

COGNIQUEST – DSA Game Based Learning App is designed as an interactive and engaging platform for learning Data Structures and Algorithms through a game-based approach. Data Structures and Algorithms are important foundations of computer science and play a major role in programming, problem-solving, technical interviews, coding competitions and placement examinations. However, many students find DSA difficult because the concepts require logical thinking, continuous practice and a clear understanding of how algorithms work. Traditional learning methods mainly focus on theoretical explanations and coding exercises, which may not always provide sufficient motivation for students to learn consistently. COGNIQUEST addresses this challenge by combining educational content with interactive game elements.

The main objective of COGNIQUEST is to make DSA learning more interesting, structured, and motivating for students. The application provides a systematic learning path through important DSA topics such as arrays, strings, linked

lists, stacks, queues, searching, sorting, recursion, trees and graphs. Instead of presenting all the concepts at once, the application allows learners to progress through different levels and missions. This approach helps students learn concepts step by step and practice them regularly. By dividing the learning process into smaller challenges, the system makes complex topics easier to approach and understand.

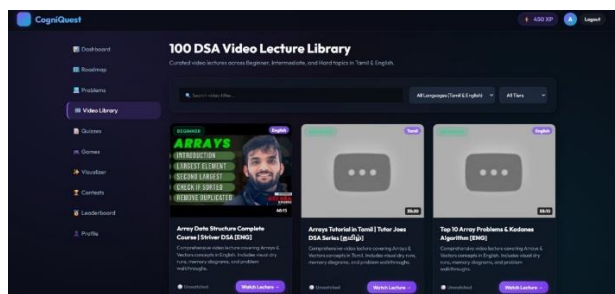
One of the major contributions of the project is the integration of gamification into DSA education. Features such as experience points (XP), levels, achievement badges, missions, rewards, lives, daily streaks and leaderboards are incorporated to encourage students to participate actively in the learning process. XP and level systems provide learners with a clear sense of progress, while badges and rewards recognize their achievements. The lives system introduces an element of challenge and streaks encourage students to maintain regular learning habits. The leaderboard can also create healthy competition among learners and motivate them to improve their performance.

The application also provides progress tracking features that allow learners to understand their learning status. Students can identify completed topics, earned XP, achieved levels, completed missions and obtained badges. This helps learners monitor their performance and recognize the areas that require additional practice. Progress tracking is especially useful for students preparing for technical interviews and placement examinations because consistent practice and revision are essential for improving DSA skills.

The proposed system also attempts to reduce the gap between theoretical knowledge and practical problem-solving. Learning DSA only through definitions and textbook examples may not be sufficient to develop strong programming skills. COGNIQUEST provides missions and challenges that encourage learners to apply the concepts they have studied. Repeated interaction with problems can improve logical thinking, algorithmic reasoning and confidence in solving programming-related questions. The game-based structure also provides an environment where learners can practice without feeling that every activity is a formal examination.

The project demonstrates that educational applications can use game-based concepts to improve the learning experience without reducing the academic value of the subject. The purpose of gamification in COGNIQUEST is not simply to make the application entertaining, but to encourage active participation, continuous practice and goal-oriented learning. The combination of learning modules, missions, rewards and progress monitoring provides a balanced environment where students can study concepts and practice them in an engaging manner.

From the technical perspective, the project provides an integrated application structure consisting of frontend, backend, database and multiple functional modules. The frontend provides the user interface through which learners interact with the application. The backend manages application logic and communication between different modules, while the database stores required user and progress-related information. Modules such as authentication, profile management, dashboard, learning content, missions, XP, achievements, rewards, leaderboard and progress tracking work together to provide a complete learning experience. This integration also demonstrates the practical application of software development concepts in an educational domain.



VIII. FUTURE WORK

The future scope of COGNIQUEST – DSA Game-Based Learning App offers numerous opportunities for enhancement through advanced technologies, personalized learning methods, interactive educational tools, and intelligent assessment mechanisms. The primary objective of future development is to transform the application into a comprehensive learning platform that helps students understand Data Structures and Algorithms (DSA) through practical implementation, interactive visualization, continuous assessment, and game-based activities.

Although the current system incorporates gamification features such as missions, experience points (XP), levels, achievement badges, rewards, lives, daily streaks, leaderboards, and progress tracking, several additional functionalities can be introduced to improve the learning experience and support students with different levels of programming knowledge. These enhancements can help make DSA learning more accessible, engaging, structured, and effective.

One of the major directions for future development is the integration of Artificial Intelligence (AI) for personalized learning. AI techniques can be used to analyze student performance, learning patterns, quiz results, completed missions, and frequently encountered mistakes. Based on this information, the application could recommend appropriate

learning materials, practice questions, and challenges according to each student's individual requirements.

For example, if a student repeatedly struggles with linked lists or recursion, the system could recommend introductory explanations, additional exercises, and guided practice before allowing the student to proceed to more advanced topics. Similarly, students who demonstrate a strong understanding of fundamental concepts could receive more complex problems to improve their analytical and programming abilities. This personalized approach would help reduce the limitations of a uniform learning path and provide a more flexible learning experience.

Another important area of future enhancement is the development of an AI-powered virtual learning assistant. This assistant could help students understand difficult DSA concepts by providing explanations, examples, hints, and step-by-step guidance. Students could ask questions about arrays, stacks, queues, linked lists, trees, graphs, searching, sorting, recursion, and other algorithmic concepts through a conversational interface. Instead of simply displaying the correct answer, the assistant could encourage students to think logically by offering progressive hints and explaining the reasoning behind a solution.

It could also provide alternative explanations when a learner does not understand the initial response. Such a feature would support independent learning and enable students to clarify doubts while practicing problems. The assistant could be further improved to recognize common misconceptions and recommend suitable learning resources based on the student's questions and performance. The application can also be enhanced through interactive visualization of data structures and algorithms. Many students find DSA difficult because the internal operations of data structures and the execution of algorithms are not always easy to understand through static diagrams or theoretical explanations. In future versions, the application could include animated representations of data structures and algorithmic operations.

For example, students could observe how elements are inserted into and removed from stacks and queues, how nodes are connected in linked lists, how trees are constructed, and how graph traversal algorithms visit different vertices. Sorting algorithms such as bubble sort, selection sort, insertion sort, merge sort, and quicksort could be demonstrated through step-by-step animations.

Searching algorithms such as linear search and binary search could also be represented visually to explain how elements are examined and how the search space changes during execution.

These visualizations would help learners connect theoretical concepts with practical implementation and develop a clearer understanding of algorithmic behavior.

Another promising direction is the implementation of an adaptive difficulty management system. In the current learning approach, students progress through structured topics and challenges. In the future, the difficulty of questions could be automatically adjusted according to individual performance. The system could consider factors such as answer accuracy, completion time, repeated attempts, hint usage, and previous assessment results to estimate the learner's current level of understanding.

Future development can also focus on introducing an integrated coding environment within the application. A built-in code editor would allow students to write, execute, debug, and test programs without switching between multiple platforms. The environment could initially support commonly used programming languages such as Python, Java, and C++, depending on implementation requirements.

Students could implement algorithms and data structures directly after completing the corresponding learning modules. Automated test cases could be used to evaluate program correctness by checking whether the output matches the expected result for different inputs.

The application could also provide meaningful feedback when a program fails a test case. By combining theoretical learning, coding practice, and automated evaluation within a single platform, COGNIQUEST could offer a more complete DSA learning experience.

In addition to code execution, the application could incorporate an intelligent error analysis and debugging mechanism. Many students experience difficulties not only in writing programs but also in identifying and correcting errors. A future version could analyze common syntax errors, logical mistakes, incorrect boundary conditions, and inefficient algorithmic approaches. Based on the detected issue, the system could explain why the program is not behaving as expected and suggest possible areas for correction.

For example, if a student implements binary search incorrectly by updating the search boundaries in the wrong order, the application could highlight the relevant logic and explain how the search interval should be modified. Such feedback would encourage students to understand the underlying problem instead of merely copying a corrected solution. The feature could also introduce debugging missions in which learners identify and repair errors in incomplete or incorrect programs.

Another significant enhancement is the inclusion of algorithm complexity analysis. Understanding time complexity and space complexity is essential for developing efficient algorithms and preparing for technical interviews. Future versions of COGNIQUEST could introduce interactive lessons on Big-O notation, best-case complexity, average-case complexity, and worst-case complexity.

Students could compare different solutions to the same problem and examine how their performance changes as the input size increases. For example, the application could compare linear search and binary search or demonstrate the differences between simple sorting algorithms and more efficient sorting techniques.

The existing gamification mechanism can be further strengthened through advanced reward and achievement systems. Future versions could introduce achievement categories based on consistency, accuracy, problem-solving improvement, topic mastery, and successful completion of advanced challenges. For example, students could earn special badges for completing all sorting challenges, solving a sequence of recursion problems, or maintaining a regular learning streak.

Additional mission types could include daily challenges, weekly goals, topic-based quests, and progressive problem-solving campaigns. The reward system could recognize not only the number of completed questions but also the effort students invest in understanding difficult concepts and improving their performance. This would help ensure that the game mechanics support educational objectives rather than encouraging students to focus exclusively on collecting points.

IX. REFERENCE

1. **Bai, S., Hew, K. F., & Huang, B. (2020).** Does gamification improve student learning outcome? Evidence from a meta-analysis and synthesis of qualitative data in educational contexts. *Educational Research Review*, 30, 100322. <https://doi.org/10.1016/j.edurev.2020.100322>
2. **Bellotti, F., Berta, R., De Gloria, A., Lavagnino, E., Antonaci, A., Dagnino, F., Ott, M., Romero, M., Usart, M., & Mayer, I. S. (2014).** Serious games and the development of an entrepreneurial mindset in higher education engineering students *Entertainment Computing*, 5(4), 357–366. <https://doi.org/10.1016/j.entcom.2014.07.003>
3. **Çakroğlu, Ü., Başbüyük, B., Güler, M., Atabay, M., & Yılmaz Memiş, B. (2017).** Gamifying an ICT course: Influences on engagement and academic performance. *Computers in Human Behavior*, 69, 98–107. <https://doi.org/10.1016/j.chb.2016.12.018>
4. **Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009).** *Introduction to Algorithms* (3rd ed.). MIT Press.
5. **Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011).** From game design elements to gamefulness: Defining gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 9–15.
6. **Domínguez, A., Saenz-de-Navarrete, J., de-Marcos, L., Fernández-**

- Sanz, L., Pagés, C., & Martínez-Herráiz, J.-J. (2013). Gamifying learning experiences: Practical implications and outcomes. *Computers & Education*, 63, 380–392.
7. Gee, J. P. (2003). *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan.
8. Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does gamification work? A literature review of empirical studies on gamification. *Proceedings of the 47th Annual Hawaii International Conference on System Sciences*, 3025–3034.
9. Hundhausen, C. D., Douglas, S. A., & Stasko, J. T. (2002). A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing*, 13(3), 259–290.
10. Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education*. Wiley.
11. Kogan, G., Chassidim, H., & Rabaev, I. (2024). The efficacy of animation and visualization in teaching data structures: A case study. *Education and Information Technologies*.
<https://doi.org/10.1007/s11423-024-10382-w>
12. Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *Proceedings of the 10th Annual SIGCSE Conference on Innovation & Technology in Computer Science Education*, 14–18.
13. Mayer, R. E. (2009). *Multimedia Learning* (2nd ed.). Cambridge University Press.
14. Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., & Velázquez-Iturbide, J. Á. (2003). Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin*, 35(2), 131–152.
15. Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.
16. Sailer, M., Hense, J. U., Mayr, S. K., & Mandl, H. (2017). How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in Human Behavior*, 69, 371–380.
17. Seaborn, K., & Fels, D. I. (2015). Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, 74, 14–31.