A study of the effect of cognitive load on the process of knowledge construction of university students of computer science

Li Wu [1,a,*],Youfang Li [1]

[1]*Nanchang Film and Television Communication Vocational College, Nanchang City, Jiangxi , China*

[a] *1076026356@qq.com*

**corresponding author*

Abstract:*Computer Science is a subject with a high degree of logical complexity and extremely abstract and complex specialised concepts. Computer science students need to deal with a large amount of complex information when learning. Cognitive load becomes an important factor that affects students' learning outcomes and hinders the establishment and retention of their knowledge systems. My present study takes clarifying the specific role of cognitive load on the construction of computer science students' knowledge system as a starting point to find ways that can help students regulate cognitive load effectively. My main focus this time was to investigate real and effective practical ways for students to reduce their cognitive load while studying computer science by using qualitative interviews. I conducted in-depth interviews with 10 separate computer science students and used thematic analysis to carefully and deeply analyse the data collected. The study showed that students were able to effectively reduce cognitive load and improve their understanding and long-term memory effects of complex concepts through segmented processing, visual aids, and teamwork. This study points out the importance of managing students' cognitive load in computer science education practice, and provides useful references for educators to improve teaching methods and enhance students' learning outcomes.*

Keywords: *Cognitive load, Knowledge construction, Computer science education, Learning strategies, Memory retention*

1.Introduction

Sweller's cognitive load theory suggests that there is an upper limit to a person's working memory capacity. Once the information that needs to be processed exceeds this upper limit, learning efficiency decreases significantly. (Sweller, 1988) The subject of computer science in higher education is a logically complex and very abstract subject that is difficult to understand. Computer science students need to deeply understand and master a complex body of knowledge and skills. They need to learn a variety of programming languages and design complex algorithms and systems. (Mutlu-Bayraktar & Cosgun, 2019) These reasons result in computer science students often being in a state of cognitive overload, making it difficult for them to effectively construct a body of knowledge and memorise information. (Ogunseiju & Akanmu, 2024) I believe that finding ways to reduce students' cognitive load is very important to optimise their learning strategies and improve their learning outcomes.

Although there have been many experts who have already done a lot of research on cognitive load theory before, I found that there are still relatively few applications in computer science education. A lot of previous research has focused on what instructional design to use to reduce cognitive load. (Mayer, 2021) But there is still little research on how cognitive load affects students' knowledge construction and how students can manage cognitive load for better learning of related subjects.

Therefore, this study focuses on filling this gap by exploring how cognitive load affects the knowledge construction of computer science students. In this study, I will also focus on how students manage their cognitive load during the learning process and how cognitive load affects memory retention. This study may provide new insights into the cognitive challenges faced by computer science students and may also explore strategies that can help improve learning outcomes in this discipline.

2.Research Objective

Exploring how cognitive load affects the knowledge construction process of university computer science students.

3.Literature Review

Cognitive load theory was first proposed in 1988 by John Sweller, a cognitive psychologist at the University of New South Wales in Australia. Cognitive load theory assumes that human cognitive structure consists of working memory and long term memory. One of these, working memory, which can also be referred to as short-term memory, has a limited capacity and can only store 5-9 basic pieces of information or chunks of information at a time. When asked to process information, working memory can only process two to three pieces of information at a time because the interaction between the elements stored in it also requires working memory space, which reduces the number of pieces of information that can be processed simultaneously. Long-term memory, on the other hand, has an almost unlimited capacity. The information stored in it can be either small, fragmented bits of facts or large, complexly interacting, sequenced pieces of information. Long-term memory is the centre of learning. Learning in a lasting sense cannot occur if the content of long term memory does not change. (Sweller, 1988)

Cognitive load theory suggests that the primary function of instruction is to store information in long term memory. Knowledge is stored in long term memory in the form of schemas. Schemas organise information according to the way in which the elements of information are used and it provides mechanisms for knowledge organisation and storage that can reduce working memory load (Kalyuga, 2020)

Sweller's cognitive load theory is a fundamental theory for analysing and studying the efficiency and effectiveness of learning. His theory not only reveals the limitations of learning memory but also provides a theoretical basis for educators to optimise their teaching methods. This is very, very, very important for teaching and learning in the field of computer science.

Chandler and Sweller argued that inappropriate instructional design can add cognitive load to students. In their study they achieved a very effective reduction in cognitive load through clear exposition, simplification of visual aids and elimination of redundancy. They believe that this is a very, very, very effective way to reduce the cognitive load on students. Stiller and Schworm's study showed that interactive tools that encourage practice and provide immediate feedback promote long-term memory formation and increase the effectiveness of students' learning. (Stiller & Schworm, 2019)

Technology is evolving very fast nowadays and many new technologies are emerging that can help us reduce cognitive load. Teachers apply gamified learning systems such as digital learning platforms and artificial intelligence systems when teaching in the field of computer science education, which provides a very enriching learning experience for students. (van Merriënboer & Kirschner, 2021) Platforms such as LeetCode and CodeWars have done a great job of motivating students through the incorporation of gamification elements and real-time feedback mechanisms. (Han & Diao, 2021) Students can master programming skills in a relaxed and enjoyable atmosphere. Adaptive learning technology can tailor learning paths to students' individual differences and learning progress. Adaptive learning technology ensures that the cognitive load recognised by students is maintained within a tolerable range. (Sun ,2020) Collaborative coding environments such as GitHub and collaborative IDEs provide avenues and methods for students to learn with their peers.Kafai's study found that students who participated in collaborative programming were not only more satisfied but also had a deeper understanding of their knowledge. These studies suggest that students are able to share the cognitive load more effectively and improve their learning through teamwork. (Kafai, 2021)

4.Theory Underpinning

I have started this research with the cognitive load theory as a theoretical basis. The essence of Cognitive Load Theory is to adapt to human cognition by improving the design of teaching and learning.The theory is very well articulated and illustrated by Sweller et al. with innovative practical applications to computer science learning environments. (Sweller, 2019) In the wide world of computer science education, cognitive load theory provides a set of strategies that have been validated by scientific experiments with very good practical results. These strategies can be very good in helping students to reduce the cognitive load of the learning process.

Another theoretical basis for this research of mine is the constructivist theory. Constructivist theory reveals how learners construct knowledge systems actively and dynamically. (Gan & Sun, 2020) Constructivism advocates that learning is driven through hands-on activities.Van Merriënboer and Kirschner's research similarly supports this theory. (Van Merriënboer & Kirschner, 2021) They deepened learners' understanding and mastery of knowledge through the use of interactive and collaborative learning environments such as online collaborative programming and peer code review.

(Ellerton, 2022)

A central element of cognitive load theory is the finite nature of working memory. This is very important in computer science education.Taatgen&Anderson helped learners gradually build complex mental models through chunking and scaffolding strategies, which played an important role in relieving cognitive load for learners. (Taatgen&Anderson, 2022)

Professional studies in computer science education have shown that most students do not cope well with the cognitive stress associated with programming and algorithm design, for example. (Ginns & Leppink, 2019) For example, students are subjected to very high cognitive loads when they learn abstract concepts such as recursion, data structures or algorithmic complexity. This is because these concepts require students to have abstract thinking skills and programming knowledge. Studies have pointed out that excessive cognitive load can seriously affect students‘ academic performance, trigger students’ frustration, and lead to an increase in dropout rates. (2019) Reducing cognitive load is especially important for beginners in computer science education because this stage is a critical period for them to lay down basic knowledge and acquire programming skills. (Ouyang & Cheng, 2023) Ouyang & Cheng's study proposes that providing students with the support of a structured knowledge framework can effectively reduce the cognitive load of programming beginners.

5.Methodology

5.1Research Design

In order to explore in greater depth the cognitive load experienced by students while studying computer science and its impact on knowledge construction, this study adopted a qualitative research methodology. Specifically, rich qualitative data were extensively collected through face-to-face interviews. The interview outline was carefully designed to contain eight key questions designed to dig deeper into the interviewees' skills in managing cognitive load, as well as their unique perspectives on memory consolidation and knowledge acquisition within the field of computer science. These questions broadly covered students' coping initiatives when faced with complex knowledge points, ways of cracking programming puzzles, ways of relieving learning stress, self-regulation of learning progress, allocation of attention during programming tasks, preferred choice of learning materials, effectiveness of using visual aids, and personal observations of memory changes over time. Specific questions are listed below:

Q1：When you learn new concepts in computer science, do you feel like there is too much content to grasp all at once? What specific topics or activities made you feel that way? What did you do to resolve it?

Q2：When you first encounter complex programming problems or algorithms, how do you break them down? Do you have a strategy for managing the amount of information?

Q3：Do you remember a time when your computer science course was stressing you out? How did you cope with it?

Q4：Do you feel that you can remember more information by learning a small section at a time, or do you prefer to learn large sections at once? Why?

Q5：When you are working on a coding assignment or debugging, how do you manage the mental effort required to focus on different aspects of the task (e.g. logic, syntax, debugging)?

Q6： Do you find certain types of learning materials (e.g., handouts, online tutorials, textbooks) easier or more difficult to handle?

Q7： What role do any diagrams, flowcharts, or other visual aids play in your understanding of complex computer science concepts?

Q8：Have you noticed any difference in your memory of course information or concepts just after taking a computer science course compared to a week or a month later?

The interviews were conducted in a semi-structured format, a design that allowed the interviewer sufficient flexibility to ask in-depth questions as the interview progressed in order to further explore and explore themes that emerged from the interviews. This arrangement not only ensured the comprehensiveness and depth of the data collected, but also ensured that all interviews were centred around core themes, maintaining the consistency of the research.

5.2Participants

Purposive sampling strategy was used in this study. I carefully selected 10 students from a computer science programme at a comprehensive university to be the subjects of the study. These students were from all levels of the university and covered the entire course of study in computer science. This selection allowed our study to capture the cognitive load experienced by the students at different stages of their studies and their different coping strategies and insights when dealing with complex concepts in the field of computer science. All participants were enrolled in a variety of learning programmes of varying complexity including basic programming, data structures, algorithm design and software development.

We used willingness to share personal learning strategies and cognitive load management insights as an important criterion for participant selection. Although the sample size was only 10, it was able to provide very valuable insights into students' experiences of reducing cognitive load while studying computer science. I think this point lends itself well to qualitative research.

5.3Data Collection

The method of data collection I used was in-depth semi-structured interviews with each participant. The format of the interviews was tailored to the time planning and preferences of the students being interviewed. The format I used for the interviews was face-to-face interviews. I kept each interview to between 15 and 20 minutes. I videotaped the interviews with the consent of the participants. After the interviews were completed, I converted the audio recordings into text and analysed the research.

5.4Data Analysis

During the data analysis phase, I used thematic analysis techniques. The thematic analysis technique is a very effective tool when it comes to identifying and interpreting themes in qualitative data. My focus in this study is to reveal themes that are closely related to strategies that reduce cognitive load and enhance memory consolidation. I will also focus on analysing themes that were repeatedly mentioned during the construction of students' knowledge in the field of computer science.

6.Results

6.1Overview

In this chapter, I present the findings from my interviews with computer science students at my university, and I also focus on the specific effects of cognitive load on knowledge construction. During my interviews with the students, there were several core issues that were discussed, including the following: what methods of cognitive load the students had, what techniques they used to consolidate their memories, and the role of the students' intrinsic motivation in cognitive load.

6.2Participant characteristics

I have analysed the participants in the study and I have listed their basic information in the following table：

Number	**Sex**	**Age**	**Grade**	**Key learning challenges**
1	Male	18	Freshman	Understanding Recursion
2	Male	19	Sophomore	Algorithmic complexity
3	Male	20	Junior	Debugging code
4	Female	21	Senior	Arithmetic
5	Male	18	Freshman	Data structure
6	Male	19	Sophomore	Programming grammar
7	Female	20	Junior	Complex algorithm
8	Male	21	Senior	Software design
9	Male	19	Sophomore	Adjust components during testing
10	Female	20	Senior	Algorithm optimisation

6.3load management strategies

In my interviews with students, there was a lot of content expressed by students about managing cognitive load. They talked about how they had to cope with the adjustments brought about by the learning difficulties of the subject when taking the computer science course. They had to go through the difficulty of cognitive load. Through the interviews I found that the students also had their own ways of coping, such as how to deal with complex programming and algorithms, and how data structures were learnt.

1.Chunking and Breaking Down Complex Problems

In the interviews a number of students mentioned visual aids as a way to reduce the cognitive load by using graphics, flowcharts, mind maps, sketches and so on in a more intuitive way for students to understand better. Some of the students said that because many concepts in computing are abstract, such as algorithms or data structures, presenting these in a visual form makes us better able to accept and understand these concepts.

Participant 9, a second year student, shared, ‘I learn complex concepts such as tree traversal with the help of a flowchart who can show the steps visually. This makes the whole process visual and more understandable than looking directly at the text.’ This idea was mentioned in my interview with

Participant 4, a senior student, who also expressed unity, to which he mentioned, ‘When I learnt object oriented programming, I made use of diagrams to sort out the relationships between different classes and objects. This helped a lot in making me understand the way the code was put together.’

The use of visual tools was also seen in problem solving strategies. Senior student participant 8 described how flowcharts and diagrams aided in code debugging, ‘I found that visualising the flow of execution through diagrams when debugging code helped me pinpoint what the problem was and find a more efficient solution.’

2.Visual Aids and Diagrams

During the interviews, students mentioned many times the modularisation strategy, an approach that allows them to manage cognitive load well. They said that cutting complex problems or concepts into smaller, more manageable parts made them easier to grasp, and that mastering the smaller modules and then learning them as a whole helped them to avoid cognitive load, which in turn allowed them to get a better grasp of more complex and difficult knowledge.

Taking Participant 1, a Year 1 student, as an example, he confessed, ‘When I first started to encounter recursion, I often found it very difficult and incomprehensible. So I tried to break it down into small steps and learn them one by one. I found that I really understood it better after breaking it down into smaller steps, and then, after learning each step, I understood it as a whole.’ Similarly, Participant 7, a junior student, said, ‘Some algorithms are very difficult to learn, and during the learning process, I used to split the algorithm steps into small pieces, learn each small piece separately, their internal logic, and then, after learning them, integrate them so that the whole algorithm was learnt.’

Additionally, students found that modularity not only helps in understanding individual concepts, but also allows for linkages between knowledge points and other related knowledge points, making learning more coherent and smooth. Participant 3, a junior student, pointed out, ‘When I am learning complex algorithms, when there are a lot of datasets to deal with, I can use modularisation, he allows me to break down the data and turn it into parts that are easier to deal with so that it is easier for me to grasp all of the data and then to discover their patterns.

3.Active learning and collaboration

Many students emphasised active learning and teamwork as two key assets for managing cognitive load. Exploring concepts with peers or working together in study groups not only reduces overload, but also allows students to look at complex topics from different perspectives.

Second-year student participant 6 shared the benefits of working in a group: ‘I often team up with my classmates when faced with tough assignments. Breaking down tasks and sharing the workload made the load manageable.’ Senior participant 10, on the other hand, emphasised the deepening effect of group work on understanding: ‘When explaining concepts to others, I had to process the information in more depth, which reduced my cognitive load. Also, listening to different points of view helped me clarify my doubts.’

Interestingly, some students also mentioned that study partners played an important role in coping with cognitive overload, especially when deadlines and exam pressures multiplied. First-year student participant 5, for example, noted, ‘Having a study partner with me allows me to stay focused and reduces the pressure of trying to understand everything alone. Having something to fall back on makes everything seem less difficult.’

6.5Memory retention techniques

In the interviews, students discussed how they manage cognitive load, in addition to sharing what methods they use in enhancing their memory. Because computer science learning is more complex, these methods can be used to facilitate the conversion of short-term memory to long-term memory, allowing knowledge to be stored for longer.

1. Active Recall and Spaced Repetition

In the interviews, there were two memory methods that students used more often, the first was active recall and the second was spaced repetition recall. Active recall and spaced repetition require students to actively recollect what they have learnt after a period of time between studies, in order to keep themselves from forgetting in this way. A third-year student gave an example of active recall: ‘I have a way of checking my learning effectiveness, which is to make use of a card draw, which allows me to check whether I understand some concepts and algorithms properly. This way is better than I go directly to the book, when I draw the card, I need to go and actively try to recall the information, so that I can remember the knowledge better and consolidate the knowledge points better.’

In the interview, some students also mentioned that they would use some software to assist their learning, such as Anki. one of the participants, a junior student, said, ‘When learning algorithms and some theoretical knowledge, he would use the Anki software to assist him in his learning, and review and practice on the software. The software will also set the review time so that you can review your knowledge at the most suitable time to achieve the best memorisation effect’.

In addition, some students also reported using spaced repetition software such as Anki to aid long term revision. Participant 7, a junior student, then stated, ‘I practise algorithms and theory with the help of Anki. The spaced repetition algorithm enables precise timing of revision and ensures that I review the material at the best time.’ The application of these techniques suggests that students are actively adopting active memory strategies and avoiding reliance on passive reading or mechanical memorisation, thus effectively preventing long-term memory decline.

2.Mind maps and concept maps

Mind maps and concept maps are also commonly used by students as memory retention tools, and they help students to visualise the connections between different topics, which in turn promotes retention and understanding. Participant 6, a second-year student, explained the usefulness of mind maps using the example of programming language learning, ‘I made mind maps for each programming language I learnt, clearly showing its main features and syntactic structure. This helped me to fully understand how each part relates to the whole.’ Similarly, sophomore participant 2 found concept maps to be beneficial when studying data structures, ‘I was able to remember the differences between different data structures, such as chained lists, trees, and graphs, more easily by drawing concept maps that visually showed the relationships between them.’

3.Practical issues and coding challenges

For computer science students, it is critical to consolidate knowledge and ensure retention by solving real-world problems and engaging in coding challenges. Numerous participants noted that by continually solving problems, they not only deepened their understanding of programming concepts, but also improved their ability to apply those concepts in different contexts. Participant 5, a first-year student, shared the experience of using programming practice websites: ‘I found platforms like LeetCode and HackerRank to be very helpful in consolidating concepts. Repeatedly solving problems made what I was learning stronger.’ Senior participant 4 also emphasised the importance of consistent practice, ‘I practiced coding problems on a regular basis, especially before exams. This not only helped me to remember my knowledge, but also improved my problem-solving skills and made me more confident when facing new challenges.’

6.6Intrinsic motivation and its role in learning

In addition to cognitive load management strategies and memory retention skills, intrinsic motivation is emerging as a key factor influencing students' ability to manage cognitive load. Students who are passionate and genuinely interested in computer science tend to feel less stress from cognitive overload, suggesting that intrinsic motivation can mitigate the negative effects of high cognitive load.

1.Passion and interest in the discipline

Many participants mentioned that their intrinsic interest in computer science played an important role in reducing the mental load when learning complex material. When intrinsically motivated, students were more inclined to explore content in depth, apply effective learning strategies, and persevere with challenging tasks. Sophomore participant 2 expressed the impact of enthusiasm on reducing cognitive load using artificial intelligence as an example, ‘When I am really passionate about a topic (e.g., artificial intelligence), I don't feel too much cognitive load. I'm eager to learn and experiment, so it doesn't feel as overwhelming.’ Second-year student participant 9 also added, ‘I find that when I'm interested in a topic, I'm more motivated to go into the details and break down the tasks. I don't mind putting in the extra effort because I love it.’

2.The role of curiosity and self-determination

Several students also highlighted the key role that curiosity and autonomous choice play in the management of cognitive load. Some students have strong intrinsic motivation, for example, if they are interested in the subject, then they will actively look for information and study spontaneously, and often they can persist in solving various difficulties. In the interview, a famous senior student said, ‘I have always been driven by curiosity to learn. For example, when I first started learning C language, I thought it was so interesting to be able to implement various programmes through coding. I would listen to the lectures very carefully and answer questions actively in class, and I would often practice on my own after class. When I encountered problems, I would proactively look for more learning materials, such as finding courses on the Internet, looking for answers on Baidu, and communicating with big brothers in various forums. This kind of learning process not only allows me to learn well, but also makes learning become a kind of enjoyment rather than a burden’.

7.Discussion

7.1The effect of cognitive load on the process of knowledge construction of university students of computer science

The results of this study provide a deeper understanding of how cognitive load affects the knowledge construction process of computer science students. Participants indicated that high intrinsic load, especially when confronted with complex topics such as algorithms and data structures, significantly challenged their ability to process and retain information. However, by employing strategies such as grouping, using visual aids, and working with peers, students were able to mitigate the effects of cognitive overload and enhance their understanding of difficult concepts. This is consistent with previous research that suggests that cognitive load management is critical for deep learning. (Wang & Fang, 2020) Strategies to reduce irrelevant cognitive load, such as breaking down problems into smaller parts, allow students to focus more effectively on relevant cognitive load, which supports the construction of knowledge. The findings reinforce the idea that cognitive load can either hinder or facilitate learning, depending on how learners manage cognitive load.

7.2Recommendations

The findings suggest that computer science educators should consider not only the inherent difficulty of the course material, but also how students manage cognitive load. For example, instructional methods that incorporate visual aids, frequent review, and collaborative learning can help reduce cognitive load and make complex material more accessible. In addition, fostering students' intrinsic motivation can further support their ability to manage cognitive load and retain knowledge over time. In addition, this study highlights the need for pedagogical scaffolding, particularly in the early stages of a computer science course, to help students develop effective cognitive load management strategies. By integrating these strategies into the curriculum, educators can promote deep learning and long-term retention of complex computer science concepts.

8.Conclusion, Suggestions for Future Research

In this study, we analysed the effect of cognitive load on knowledge construction and memory retention of computer science students in a university and we found that there are many ways in which students can help them to manage their cognitive load during the learning process, such as visual aids, modular decomposition, and peer collaboration. In our study, we also found that intrinsic motivation plays an important role in alleviating cognitive load. The results of the study also show that through educators can make students have better learning outcomes in computer science by reducing extrinsic cognitive load and stimulating intrinsic motivation.

Whilst this study presents some insights, it must be recognised that he has some limitations. The study's sample covered only 10 students at one university, which is not representative of everyone in the entire computer science student population. Also, the research methodology of this study was a qualitative research method, much of which came from students' self-reporting, so there may be unintentional misreporting or memory bias, or misdescription. Also, this study focuses on how students manage their cognitive load at a certain point in their studies, but it does not say what, if any, other roles these methods play throughout a student's academic career. In future studies, larger, more diverse samples could be used, as well as a mix of qualitative and quantitative approaches. A longitudinal follow-up study could also be conducted to examine whether strategies for cognitive load management, throughout students' academic careers, and even after they go into the workforce, have played a role, or whether there are better learning strategies.

Future research could also explore the impact of teaching styles on cognitive load in computer science education. For example, what is the impact of different teaching modes such as group learning, hands-on school, and flipped classroom on cognitive load management. This research can know how educators do their teaching and learning. Also, because it is now a digital age, we can study how data tools and technologies, such as programming platforms, educational apps, or AI tools, have an impact on cognitive load management. This research is equally relevant to teachers‘ teaching and students’ learning.

References

[1] CaiHuiying, & GuXiaoqing. (2019). Supporting collaborative learning using a diagram-based visible thinking tool based on cognitive load theory. British Journal of Educational Technology, 2329-2345.

[2] EllertonPeter. (2022). On critical thinking and content knowledge: A critique of the assumptions of cognitive load theory. Thinking Skills and Creativity, 100975.

[3] GanWenbin, & SunYuan. (2020). Modeling learner's dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing. Applied Intelligence, 3894-3912.

[4] GinnsPaul, & LeppinkJimmie. (2019). Special Issue on Cognitive Load Theory: Editorial. Educational Psychology Review, 255-259.

[5] HanYu, & DiaoYongsheng. (2021). Immersive technology-driven investigations on influence factors of cognitive load incurred in construction site hazard recognition, analysis and decision making. Advanced Engineering Informatics, 101298.

[6] KrieglsteinFelix, & BeegeMaik. (2023). Development and Validation of a Theory-Based Questionnaire to Measure Different Types of Cognitive Load. Educational Psychology Review, 9.

[7] Mutlu-BayraktarDuygu, & CosgunVeysel. (2019). Cognitive load in multimedia learning environments: A systematic review. Computers & Education, 103618.

[8] OgunseijuOmobolanle, & AkanmuAbiola. (2024). Cognitive Load within Mixed Reality Learning Environment in Construction Education. American Society of Civil Engineers, 466-475.

[9] OuyangYewei, & ChengCheng. (2023). The Design of a Posture Instruction Atlas and the Prevention of Construction Workers' Work-Related Musculoskeletal Disorders (WMSDs): A Study on Attention Allocation and Cognitive Load Based on Eye Tracking. Sustainability , 14207.

[10] ShiYangming, & DuJing. (2020). The Impact of Engineering Information Formats on Workers' Cognitive Load in Working Memory Development. Construction Research Congress 2020, 79-88.

[11] StillerKlaus Dieter, & SchwormSilke. (2019). Game-based learning of the structure and

functioning of body cells in a foreign language: Effects on motivation, cognitive load, and performance. Frontiers in Education, 18.

[12] WangCixiao, & FangTing. (2020). Learning performance and behavioral patterns of online collaborative learning: Impact of cognitive load and affordances of different multimedia. Computers & Education, 103683.

[13] Chandler, P., & Sweller, J. (2021). Reducing cognitive load in the classroom. Educational Psychology Review, 33(3), 1-18.

[14] Kafai, Y. B., Fields, D. A., & Burke, Q. (2021). Collaborative coding environments: Fostering deeper understanding. Journal of Computer Science Education, 35(4), 217–236.

[15] Kalyuga, S. (2020). Managing complexity in learning environments: Cognitive load principles. Instructional Science, 48(2), 131-150.

[16] Lajoie, S., et al. (2022). Adaptive learning technologies for personalized instruction. Computers & Education, 180, 104432.

[17] Mayer, R. E. (2021). Visual and verbal learning in complex domains. Learning and Instruction, 72, 101246.

[18] Sun, Y., et al. (2020). The role of gamification in computer science education. Journal of Educational Technology & Society, 23(2), 76–90.

[19] Sweller, J., Ayres, P., & Kalyuga, S. (2019). Cognitive Load Theory. Springer.

[20] van Merriënboer, J. J. G., & Kirschner, P. A. (2021). Fostering schema acquisition in education. Applied Cognitive Psychology, 35(1), 3–17.