Multidisciplinary Sciences, Volume 16. (2026) No. 4, pp. 3–14. *https://doi.org/10.35925/j.multi.2026.4.1*

INTEGRATING INTERACTIVE, GAMIFIED, AND GAME-BASED LEARNING FOR TEACHING PROGRAMMING

Martin Balogh[*], Erika Baksáné Varga[**‡]

University of Miskolc, 3515 Miskolc, Hungary
‡ *Corresponding author*
* baloghmartin2004@gmail.com
** erika.b.varga@uni-miskolc.hu

Abstract. Developing practical programming skills requires continuous practice beyond scheduled classroom activities, while maintaining student engagement in self-paced learning environments remains a significant challenge. This paper introduces EduTower, an integrated interactive learning environment designed to support independent practice in object-oriented programming. Previous research has identified focused learning materials, interactivity, immediate feedback, rewards, progress tracking, gamification, and game-based learning as approaches that can support student engagement. Based on these findings, the EduTower concept was designed by integrating these approaches into a coherent learning process and implementing it within Moodle. Each learning unit provides a concise theoretical summary and glossary support followed by interactive H5P activities, formative quizzes, and practical programming exercises. Selected programming tasks are integrated into Code Siege, a browser-based educational game in which successful solutions contribute directly to game progression. Student achievements are recorded in the Moodle gradebook, providing feedback and making progress visible throughout the course. The resulting pilot environment demonstrates how these complementary educational approaches can be systematically integrated to support practical C# programming and self-paced learning outside scheduled classes. The present study focuses on the design and pilot implementation of the EduTower concept.

Keywords: Computer Science Education, Programming, Educational Technology, Gamification, Game-Based Learning.

1 Introduction

Learning object-oriented programming requires not only an understanding of theoretical concepts but also continuous practical experience. Although classroom instruction provides the foundations of programming knowledge, students need opportunities to practise independently, experiment with code, identify their mistakes, and gradually develop their problem-solving skills. Providing an effective environment for self-paced practice outside scheduled classes is therefore an important complement to conventional programming instruction.

Digital learning environments can support this process by combining interactive learning activities, formative assessment, and immediate feedback. In addition to providing practice opportunities, such environments can help students monitor their progress and maintain motivation through gamified activities and game-based learning. For programming education in particular, automated feedback can support an iterative learning process in which students attempt a solution, receive information about errors, revise their code, and try again.

This paper presents the design and implementation of an interactive learning path for teaching object-oriented programming, based on the EduTower educational concept. The proposed environment integrates Moodle, H5P-based interactive activities, gamification, formative assessment, programming exercises, and the Code Siege

game into a coherent learning process. The primary objective is to provide students with a structured environment for independent practice that offers continuous feedback, supports progress monitoring, and encourages them to advance through the learning materials.

2 Related Works

2.1 Moodle and Interactive Learning with H5P

Moodle is a widely adopted open-source learning management system (LMS) in higher education that supports the organisation of learning materials, assessment, communication, and collaborative learning within a common digital environment. A systematic review of 155 studies by Gamage et al. [1] showed that the use of Moodle has evolved beyond basic content delivery towards more diverse pedagogical applications, including collaborative and adaptive learning and online assessment.

This ability to bring different learning activities together within a common environment is particularly relevant for the design of integrated learning experiences. Rather than using digital resources as isolated components, technology-enhanced and blended learning environments can combine different forms of instruction and learner interaction within a coherent learning process. Recent systematic evidence indicates that appropriately designed blended learning environments can positively influence multiple dimensions of student engagement [2]. Moodle therefore provides a suitable technological foundation for integrating different forms of digital and interactive learning.

One approach to extending Moodle beyond conventional content delivery is the integration of H5P (HTML5 Package), an open framework for creating reusable interactive learning content. H5P supports a wide variety of content types, including interactive videos, quizzes, flashcards, and branching scenarios, allowing conventional digital learning materials to be complemented with learner interaction, self-assessment, and immediate feedback.

Recent research has demonstrated the potential of H5P for supporting interactive learning in higher education. A systematic literature review by Ki [3], covering 30 peer-reviewed studies, identified student engagement, learning outcomes, and innovative pedagogical approaches as the main areas in which H5P has been investigated. The review indicates that H5P can support student engagement through interactive content, self-paced learning, and interactive assessment, while also emphasising that its use should be aligned with pedagogical objectives. Empirical evidence also supports the integration of H5P into blended learning environments. Rahmi et al. [4] evaluated H5P-based interactive content integrated into Moodle within a rotation-model blended learning setting and reported positive effects on students' knowledge, skills, and satisfaction.

However, the educational value of H5P depends not only on the availability of interactive content types but also on how these activities are integrated into the learning process. Noutcha and Kieffer [5] highlighted the lack of guidelines for designing and evaluating H5P-based online learning experiences and proposed recommendations for the systematic integration of H5P activities into blended learning. Their findings also indicate that learners value the interactive features of H5P, while content and usability remain central aspects of the learning experience. These studies suggest that the pedagogical potential of H5P lies not simply in introducing individual interactive activities, but in integrating them purposefully into a coherent learning process.

2.2 Gamification and Game-Based Learning

Gamification and game-based learning have become widely used approaches for promoting active participation in technology-enhanced education. While gamification introduces game-design elements into non-game learning activities, game-based learning incorporates actual games or game environments into the learning process [6]. Both approaches aim to increase learner motivation and engagement, but differ in how game mechanisms are integrated into instructional activities.

Recent research suggests that both gamification and game-based learning can positively influence students' learning and motivational outcomes. Gamification has been shown to enhance learning and motivation, although its effectiveness depends on the instructional context and the meaningful alignment of game elements with

pedagogical objectives [7, 8]. Similarly, game-based learning has been associated with increased learning, engagement, and motivation [9]. A recent comparative analysis of gamification and serious games further indicates that both approaches can contribute to learning achievement and motivation, while their effects may vary across different motivational dimensions [10].

Current evidence therefore suggests that neither gamification nor game-based learning should be regarded simply as a means of adding entertainment to digital learning. Their educational value depends on purposeful instructional design and the integration of game mechanisms with learning objectives [11]. This is particularly relevant to self-paced learning, where interactive and game-related elements can provide additional incentives for students to continue engaging with learning activities outside scheduled classes.

2.3 Interactive and Game-Based Approaches in Programming Education

Programming education has increasingly adopted interactive, gamified, and game-based approaches to address challenges related to student motivation, engagement, and the acquisition of practical programming skills. Similar trends can be observed more broadly in technology-enhanced computer science education, where interactive and simulation-based applications have been used to connect theoretical concepts with practical experience and to increase learner engagement. For example, Drămnesc et al. [12] presented AI-based robotic applications that combine interactive learning assistants and simulation-based educational use cases to make complex theoretical topics more accessible and engaging. A systematic review by Ishaq and Alvi [13], covering 81 studies, indicates that gamification-based approaches can improve motivation, engagement, and learning outcomes in programming courses, although their effectiveness depends on the selected game elements, learner characteristics, and instructional design.

These approaches have also been investigated specifically in object-oriented programming (OOP) education, where the abstract nature of concepts such as classes, objects, inheritance, and polymorphism can create difficulties for novice learners. Jusas et al. [14] incorporated gamification and interactive content into a Moodle-based OOP course using experience points, interactive activities, and collaborative game elements, and reported increased student engagement and improved learning outcomes. More explicitly game-based approaches have also been developed for OOP education. Djelil and Sanchez [15], for example, introduced Progo, a game-based learning environment specifically designed to support beginners in learning OOP. Their work emphasises that effective game-based learning requires the meaningful integration of programming knowledge into the game experience rather than the simple addition of game elements to conventional instruction. Sunday et al. [16] evaluated Imikode, a virtual reality educational game designed to introduce OOP concepts to novice learners. The environment combines game-based learning with an intelligent tutoring component and AI-based real-time error feedback, and students reported high levels of satisfaction with its use.

Automated feedback is another important component of interactive programming education, as it enables students to repeatedly submit, evaluate, and revise their solutions without requiring continuous instructor intervention. Previous research has shown that automated feedback systems are widely used in programming education to provide immediate information about errors and solution correctness [17].

2.4 Research Objective

Previous research has demonstrated the potential of interactive learning, gamification, game-based learning, and automated feedback in programming education. However, these approaches have predominantly been investigated individually or in various combinations, while our literature review did not identify an integrated environment that brings them together into a coherent learning path for object-oriented programming.

To address this gap, this study introduces the EduTower concept, which integrates established engagement-supporting approaches within a unified learning environment for practical OOP education. The aim of the study is to present the design and pilot implementation of EduTower as a structured environment for independent practice outside scheduled classes.

3 Methodology

This study follows a design- and implementation-oriented approach. Based on findings identified in the literature review, pedagogical and technological components associated with learner engagement were selected and systematically integrated into the EduTower concept. The resulting concept was then implemented and instantiated in a C# object-oriented programming practical course.

3.1 Instructional Design Principles

The design of the proposed learning environment is grounded in established pedagogical principles for technology-enhanced learning. Systematic instructional design plays an important role in the development of effective and consistent digital learning environments, as it provides a structured basis for aligning learning objectives, instructional activities, assessment, and technological resources [18]. Accordingly, the course was developed to support active student participation by combining theoretical instruction, interactive learning activities, formative assessment, and game-based programming exercises within a unified learning environment.

The overall instructional approach follows the principles of constructivist learning, which views learning as an active process in which students construct knowledge through meaningful learning activities rather than passively receiving information [19, 20]. Interactive tasks, programming exercises, and problem-solving activities therefore play a central role throughout the course rather than serving only as supplementary assessment components.

To increase learner motivation and engagement, the course incorporates principles of gamification and game-based learning, with game elements aligned with the instructional objectives [6, 7, 11].

Another important design principle is the continuous provision of formative assessment and immediate feedback. Interactive H5P activities enable learners to check their understanding throughout the learning process, while immediate feedback helps them identify misconceptions before progressing to more complex programming tasks. Recent reviews emphasize that timely, formative feedback is one of the key factors supporting effective digital learning environments and learner engagement [21, 22].

3.2 The EduTower Educational Concept

Based on the instructional design principles described above, the developed learning environment follows the EduTower educational concept proposed in this work. EduTower integrates interactive learning, gamification, game-based learning, formative assessment, and automated feedback within a unified educational ecosystem designed for teaching object-oriented programming. Rather than treating these approaches as independent elements, the concept connects them into a coherent learning process that supports individual, self-paced learning.

The EduTower concept consists of two interconnected learning environments. The first is a Moodle-based environment that provides theoretical materials, interactive H5P activities, quizzes, and programming exercises. A point-based assessment mechanism introduces gamification into this environment: students earn points by completing learning activities, and their results are recorded in the Moodle gradebook. The accumulated results provide learners with an overview of their performance and enable them to monitor and self-assess their progress.

The second environment is the custom-developed Code Siege educational game, which extends the learning process through game-based programming challenges. Students apply the OOP concepts introduced in the Moodle course by solving C# coding tasks within an actual game environment. Submitted solutions are automatically evaluated, and feedback is provided when errors are detected, allowing students to revise their code and attempt the challenge again. Successful solutions enable progression within the game.

Formative assessment and automated feedback therefore operate as cross-cutting mechanisms across the EduTower learning process. At the Moodle/H5P level, students receive immediate feedback on interactive activities and quizzes, while at the Code Siege level, feedback is based on the evaluation of submitted programming solutions. The Moodle gradebook provides a central record of achievement, bringing together

students' results and supporting continuous progress monitoring. In this way, EduTower establishes a continuous progression from theoretical knowledge acquisition through interactive practice and self-assessment to practical programming and game-based application of the obtained skills.

3.3 Course Design

Within the EduTower ecosystem, the Moodle course implements the instructional component of the proposed concept. The course content is organised into consecutive learning modules covering the fundamental topics of object-oriented programming, including classes and objects, encapsulation, inheritance, polymorphism, interfaces, and advanced programming concepts. The modules gradually increase in complexity while maintaining a consistent instructional structure throughout the course.

The modular organisation of the course facilitates future extension and maintenance. New learning modules, H5P activities, programming exercises, and game levels can be incorporated without modifying the overall course organisation, making the learning environment readily adaptable to future curriculum development.

4 Results

4.1 Overview of the Implemented Learning Environment

A complete interactive learning environment for teaching object-oriented programming in C# was successfully developed and deployed in the Moodle LMS of the University of Miskolc with Hungarian user interface and is currently available only to enrolled students. The course follows a structured learning path presented in Figure 1. Each topic begins with concise theoretical material, followed by glossary-supported explanations of key concepts. Students then complete one or more interactive H5P activities before taking a formative quiz to assess their understanding. Finally, practical programming exercises reinforce the newly acquired knowledge. The final implementation consists of:

- 10 learning modules covering the main topics of object-oriented programming;
- 10 of each H5P interactive activity (flashcard sets, crossword activities, branching scenarios, formative quizzes, code-completion exercises and interactive videos);
- 65 glossary entries;
- 3 educational game levels implemented in the Code Siege game.

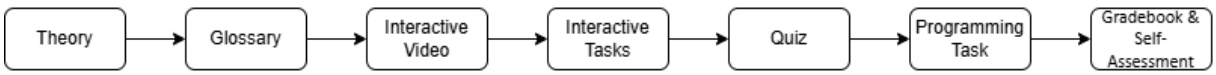


Figure 1. *Overall structure of the interactive learning path.*

Overall, the learning environment contains approximately 12 minutes of instructional video material and 67 interactive assessment items distributed throughout the course.

4.2 Implemented H5P Activities

The learning path integrates several H5P content types, each designed to support a specific learning objective as summarized in Table 1.

Table 1

H5P activity types and their educational purposes.

Activity Type	Learning Objectives
Interactive videos	Practical demonstration, code tracing, understanding program execution
Flashcards	Recall of programming concepts and syntax
Crosswords	Reinforcement of terminology

Activity Type	**Learning Objectives**
Branching scenarios	Problem solving, programming decisions, comparison of alternative solutions
Quizzes	Code comprehension, error identification, formative self-assessment
Code-completion exercises	Code construction, application of programming syntax and logic

Interactive videos introduce theoretical concepts using narrated code demonstrations. Figure 2 illustrates the application of embedded checkpoints that require learners to answer questions before continuing the video, thereby encouraging active rather than passive learning.

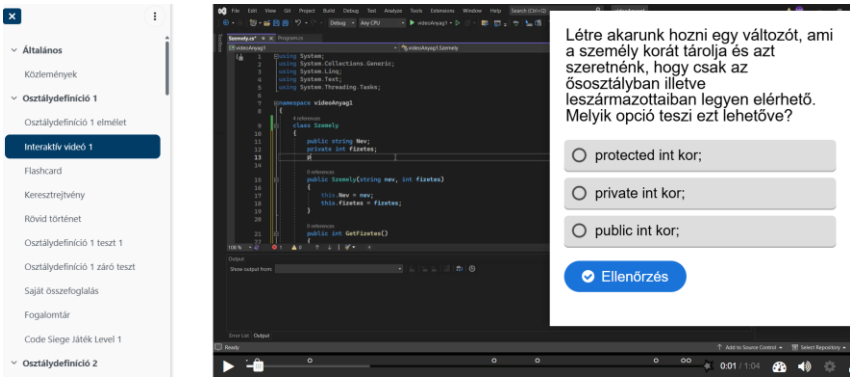


Figure 2. *Interactive H5P video with an embedded comprehension question.*

Flashcards reinforce programming terminology and object-oriented concepts through self-paced repetition and immediate feedback, while crossword activities provide an additional mechanism for consolidating technical vocabulary in an engaging format. Figure 3 shows examples for both activities.

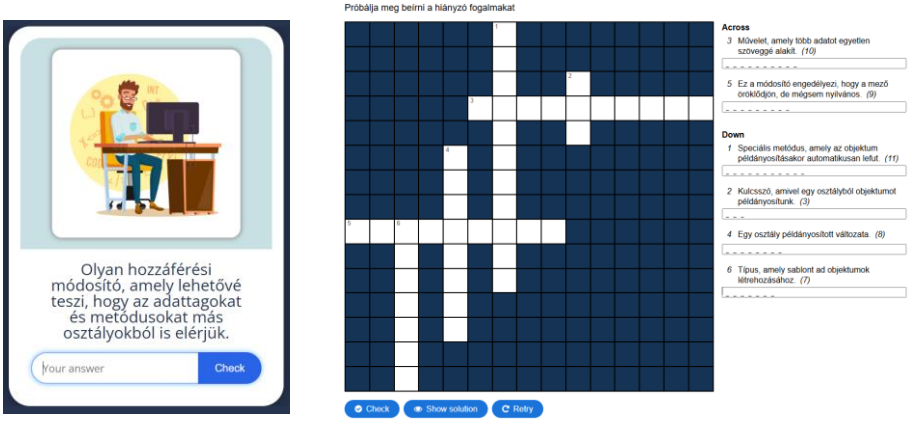


Figure 3. *H5P flashcard and crossword activities supporting terminology reinforcement.*

Branching Scenario activities simulate programming-related decision-making situations illustrated in Figure 4. Students navigate alternative solution paths based on their choices, allowing them to explore the consequences of different software design decisions.

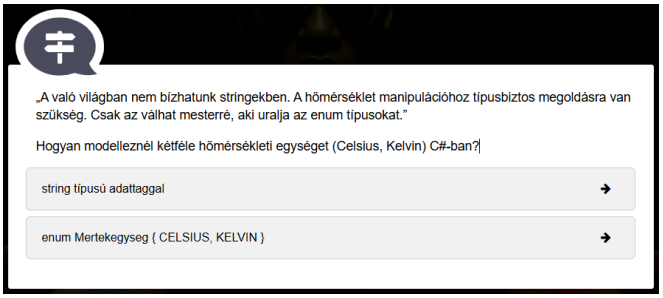


Figure 4. *Example of an H5P branching scenario used for programming-related decision making.*

Formative quizzes combine multiple-choice questions, short-answer questions, and code-completion tasks to assess conceptual understanding. An example is shown in Figure 5. Longer programming exercises require students to complete partially implemented C# code fragments, supporting the transition from theoretical knowledge to practical coding skills.

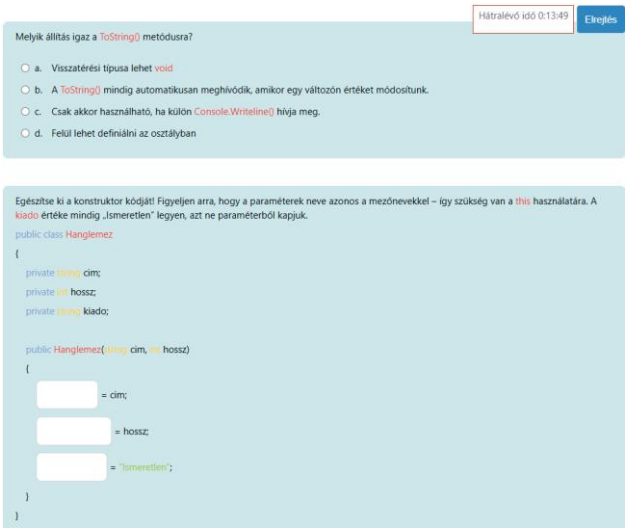


Figure 5. *H5P quiz with code-completion task.*

To facilitate continuous learning, theoretical pages are linked directly to the Moodle glossary. Important programming terms can therefore be accessed without interrupting the learning process.

4.3 Code Siege Game

Code Siege is a custom-developed Unity-based educational game that extends the EduTower learning path with game-based programming practice. The game runs in a web browser using WebGL and is launched directly from the Moodle course, allowing students to move from the learning materials to the game environment without leaving the course workflow. After completing the game activities, the score achieved is automatically synchronized with the Moodle gradebook, where it becomes part of the student's overall record of progress.

Within Code Siege, the student takes the role of an apprentice and solves C# programming challenges presented by a virtual tutor as shown in Figure 6.



Figure 6. *Programming challenge presentation and coding panel in Code Siege.*

Immediate feedback is also maintained within the game-based environment. Submitted solutions are automatically processed and evaluated. If syntax or solution errors are detected, the tutor provides feedback (Figure 7) that helps students identify the problem, revise their code, and attempt the challenge again.



Figure 7. *Example for tutor feedback in Code Siege.*

Once a correct solution has been submitted, its successful completion is translated into an in-game consequence: the student is awarded a given number of knights depending on the difficulty of the programming task. These knights are added to the student's army and become available for the subsequent siege battle.

4.4 Learning Path Integration

The individual learning activities described above were integrated into a consistent instructional sequence across the course modules, as illustrated in Figure 1. The learning path is designed primarily for self-paced individual learning, allowing students to progress through the course materials and assess their understanding at their own pace. Table 2 shows how this general learning path was instantiated for the individual OOP topics and how selected modules were connected to Code Siege.

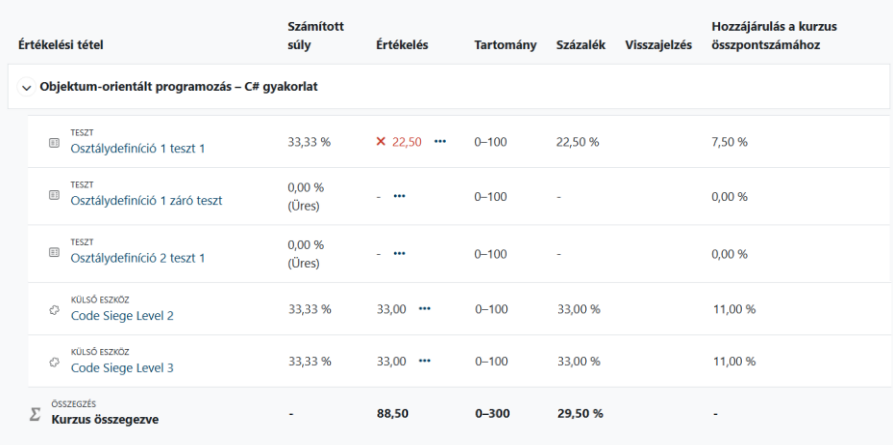
Table 2
Structure of the implemented interactive learning path.

Course topic	H5P Activities	Assessment	Code Siege Integration
Classes and Objects, Encapsulation	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	Level 1
Information Hiding	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	-
Constructor, Method Overloading	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	-
Inheritance, Method Overriding, Reference Types	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	Level 2
Inheritance, Abstract Class	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	-
Interface, Interface Reference	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	-
Nested Class, Enum, Sorting Object Arrays	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	Level 3
Exception Handling, File Handling	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	-
Relationships between Classes, UML Class Diagram	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	-
Classes of the system Namespace	Flashcard, Crossword, Branching scenario, Interactive video	Quiz, Code completion task	-

Formative assessment, immediate feedback and gamification are embedded throughout the learning path. Each lesson is followed by quizzes and interactive H5P activities that allow students to check their understanding of the presented concepts. Students receive immediate feedback on the correctness of their responses and earn

points for completing these activities. The scores achieved are automatically recorded in the Moodle gradebook, together with the scores earned in the Code Siege game (Figure 8). In this way the gradebook provides an aggregated overview of performance and enables students to monitor and self-assess their progress throughout the course.

Together, these components establish continuous progression from knowledge acquisition and interactive practice to self-assessment, programming application, and game-based learning. Moodle serves as the central environment for this process, since results from quizzes, H5P activities and the Code Siege game are recorded in the gradebook. Consequently, learner progress across both the gamified learning activities and the game-based environment can be managed within the same learning ecosystem.



Értékelési tétel	Számított súly	Értékelés	Tartomány	Százalék	Visszajelzés	Hozzájárulás a kurzus összpontszámához
Objektum-orientált programozás – C# gyakorlat						
TESZT Osztálydefiníció 1 teszt 1	33,33 %	✕ 22,50 •••	0–100	22,50 %		7,50 %
TESZT Osztálydefiníció 1 záró teszt	0,00 % (Üres)	- •••	0–100	-		0,00 %
TESZT Osztálydefiníció 2 teszt 1	0,00 % (Üres)	- •••	0–100	-		0,00 %
KÜLSŐ ESZKÖZ Code Siege Level 2	33,33 %	33,00 •••	0–100	33,00 %		11,00 %
KÜLSŐ ESZKÖZ Code Siege Level 3	33,33 %	33,00 •••	0–100	33,00 %		11,00 %
Σ ÖSSZEGZÉS **Kurzus összegezve**	-	**88,50**	**0–300**	**29,50 %**		-

Figure 8. *Excerpt from the Moodle gradebook.*

5 Discussion

The main contribution of this work is the EduTower concept, which integrates several approaches previously identified as beneficial for student engagement into a unified learning environment for practical C# programming education. Rather than introducing a new instructional technique, the concept builds on existing evidence concerning the value of focused learning materials, interactivity, immediate feedback and rewards, visible progress, and game-based learning.

These elements serve complementary roles within the proposed environment. Concise theoretical summaries and glossary links highlight the essential knowledge required for practical work, while interactive activities and quizzes encourage active participation and provide immediate feedback. Points and progress tracking make achievement visible, whereas Code Siege extends programming practice into a game-based context. Their integration into a coherent learning path is particularly relevant for self-paced learning outside scheduled classes, where maintaining student engagement without continuous instructor presence represents an important challenge.

The contribution of EduTower therefore lies in bringing these previously studied approaches together and applying them systematically to the practical teaching of object-oriented programming in C#.

6 Limitations

The main limitation of the present study is that it focuses on the design and implementation of the EduTower concept rather than its empirical evaluation. Although the individual components were selected based on previous research indicating their potential to support engagement and learning, the present study does not demonstrate that their integration increases student engagement or improves programming performance.

Accordingly, EduTower should currently be regarded as an implemented educational framework whose expected benefits require empirical validation.

A further limitation is that the current implementation was developed for a single OOP course focusing on C# within one institutional context.

7 Conclusion

This paper introduced EduTower, an integrated interactive learning environment designed to support practical C# object-oriented programming education and independent practice outside scheduled classes. The concept brings together focused theoretical support, interactive learning activities, immediate feedback, rewards, progress tracking, and game-based programming practice within a unified Moodle-based learning path.

The novelty of EduTower lies primarily in this integration rather than in the individual components themselves. By combining approaches that previous research has associated with student engagement, the proposed environment is designed to sustain students' interest while guiding them from essential theoretical concepts toward active practice and application. The implementation demonstrates the feasibility of applying this integrated concept to C# programming education, while future empirical evaluation will determine its actual effects on student engagement and learning outcomes.

Author Contributions. Conceptualization, B.V.E.; methodology, B.V.E.; software, B.M.; validation, B.V.E. and B.M.; resources, B.V.E.; writing, original draft preparation, B.V.E. and B.M.; writing, review and editing, B.V.E.; visualization, B.M.; supervision, B.V.E. All authors have read and agreed to the published version of the manuscript.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

AI Use Statement. During the preparation of this manuscript, the authors used ChatGPT (OpenAI) for language improvement, text organization, and editorial assistance. All generated content was carefully reviewed, revised, and verified by the authors. The authors take full responsibility for the accuracy and integrity of the manuscript.

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Explanation
AI	Artificial Intelligence
LMS	Learning Management System
OOP	Object Oriented Programming

References

[1] Gamage, S.H.P.W.; Ayres, J.R.; Behrend, M.B. A systematic review on trends in using Moodle for teaching and learning. *Int. Journal of STEM Education*, 2022, Vol. 9, Article no. 9. https://doi.org/10.1186/s40594-021-00323-x

[2] De Bruijn-Smolders, M.; Prinsen, F.R. Effective student engagement with blended learning: A systematic review. *Heliyon*, 2024, 10(23), e39439, ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2024.e39439

[3] Ki, Y.P. Enhancing Student Engagement and Learning Outcomes in Higher Education using H5P Interactive Learning Tools: A Systematic Literature Review. *Int. Journal of Modern Education*, 2024, 7(24), 969-990. https://doi.org/10.35631/IJMOE.724069

[4] Rahmi, U.; Ramadhani Fajri, B.; Azrul, A. Effectiveness of Interactive Content with H5P for Moodle-Learning Management System in Blended Learning. *Journal of Learning for Development*, 2024, 11(1), 66-81. https://doi.org/10.56059/jl4d.v11i1.1135

[5] Noutcha, M.; Kieffer, S. Blended Learning Based on H5P Interactive Exercises: Insights from a Case Study. In: Zaphiris, P., Ioannou, A. (eds) *Learning and Collaboration Technologies*, HCII 2024, Lecture Notes in Computer Science, vol 14722. Springer, Cham. https://doi.org/10.1007/978-3-031-61672-3_14

[6] Krath, J.; Schürmann, L.; von Korflesch, H.F.O. Revealing the theoretical basis of gamification: A systematic review and analysis of theory in research on gamification, serious games and game-based learning. *Computers in Human Behavior*, 2021, Volume 125, 106963, ISSN 0747-5632. https://doi.org/10.1016/j.chb.2021.106963

[7] Li, L.; Hew, K.F.; Du, J. Gamification enhances student intrinsic motivation, perceptions of autonomy and relatedness, but minimal impact on competency: a meta-analysis and systematic review. *Education Tech Research Dev*, 2024, Vol.72, 765-796. https://doi.org/10.1007/s11423-023-10337-7

[8] Li, M.; Ma, S.; Shi, Y. Examining the effectiveness of gamification as a tool promoting teaching and learning in educational settings: a meta-analysis. *Front. Psychol.*, 2023, 14:1253549. https://doi.org/10.3389/fpsyg.2023.1253549

[9] Mikrouli, P.; Tzafilkou, K.; Protogeros, N. Applications and Learning Outcomes of Game Based Learning in Education. *International Educational Review*, 2024, 2(1), 25-54. https://doi.org/10.58693/ier.212

[10] Ren, J.; Xu, W.; Liu, Z. The Impact of Educational Games on Learning Outcomes. *International Journal of Game-Based Learning*, 2024, Volume 14, Issue 1, ISSN 2155-6849. https://doi.org/10.4018/IJGBL.336478

[11] Hong, Y.; Saab, N.; Admiraal, W. Approaches and game elements used to tailor digital gamification for learning: A systematic literature review. *Computers & Education*, 2024, Volume 212, 105000, ISSN 0360-1315. https://doi.org/10.1016/j.compedu.2024.105000

[12] Drămnesc, I.; Ábrahám, E.; Antal, L.; Csóka, P.; Fachantidis, N.; Jebelean, T.; Kusper, G.; Papadopoulos, K. Artificial Intelligence based robotic applications for higher education. *Annales Mathematicae et Informaticae*, 2025, 61, 68-79. https://doi.org/10.33039/ami.2025.10.009

[13] Ishaq, K.; Alvi, A. Personalization, Cognition, and Gamification-based Programming Language Learning: A State-of-the-Art Systematic Literature Review. ArXiv, abs/2309.12362, 2023. https://arxiv.org/abs/2309.12362

[14] Jusas, V.; Barisas, D.; Jančiukas, M. Game Elements towards More Sustainable Learning in Object-Oriented Programming Course. *Sustainability*, 2022, 14(4), 2325. https://doi.org/10.3390/su14042325

[15] Djelil, F.; Sanchez, E. Game design and didactic transposition of knowledge. The case of Progo, a game dedicated to learning object-oriented programming. *Education and Information Technologies*, 2022, 28(1), 283-302. https://doi.org/10.1007/s10639-022-11158-6

[16] Sunday, K.; Oyelere, S.S.; Agbo, F.J.; Aliyu, M.B.; Balogun, O.S.; Bouali, N. Usability Evaluation of Imikode Virtual Reality Game to Facilitate Learning of Object-Oriented Programming. *Tech Know Learn*, 2023, 28, 1871-1902. https://doi.org/10.1007/s10758-022-09634-6

[17] Keuning, H.; Jeuring, J.; Heeren, B. A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. *ACM Trans. Comput. Educ.*, 2018, 19(1), Article 3. https://doi.org/10.1145/3231711

[18] Senadheera, V. V.; Ediriweera, D. S.; Rupasinghe, T. P. Instructional Design Models for Digital Learning in Higher Education - A Scoping Review. *Journal of Learning for Development*, 2024, 11(1), 15-26. https://doi.org/10.56059/jl4d.v11i1.973

[19] Jonassen, D. Designing Constructivist Learning Environments, In *Instructional-design Theories and Models: A New Paradigm of Instructional Theory*, Volume II, 1st ed.; Reigeluth, C.M. Ed.; Publisher: Routledge, 1999. https://doi.org/10.4324/9781410603784

[20] Seraji, F.; Musavi, H.O. Does applying the principles of constructivism learning add to the popularity of serious games? A systematic mixed studies review. *Entertainment Computing*, 2023, Vol. 47, 100585. https://doi.org/10.1016/j.entcom.2023.100585

[21] Gutierrez, R.; Govea, J.; Valencia-Silva, E. The Impact of AI-Driven Formative Feedback in Gamified Learning Environments: A Systematic Literature Review. *Education Sciences*, 2026, 16, 858. https://doi.org/10.3390/educsci16060858

[22] Hattie, J.; Timperley, H. The Power of Feedback. *Review of Educational Research*, 2007, 77(1). https://doi.org/10.3102/003465430298487