Peertechz

Trends in Computer Science and Information Technology

OPEN ACCESS JOURNAL

ISSN: 2641-3086 Journal DOI: https://dx.doi.org/10.17352/tcsit

Research Article

Programming with Scratch and Unity in Continuing Education: An Approach Featuring Simple Video Game Creation Activities

José Manuel Sáez López[1]* and Redondo-Duarte Sara[2]

[1]Associated Professor at the National University of Distance Education (UNED), Spain

[2]Assistant Professor at the National University of Distance Education (UNED), Spain

Submitted : 01 July, 2026
Accepted : 13 July, 2026
Published : 14 July, 2026

***Corresponding author:** José Manuel Sáez López, Associated Professor at the National University of Distance Education (UNED), Spain,
E-mail: jmsaezlopez@edu.uned.es

Keywords: Higher education; Programming, Teaching and learning strategies, Educational technology

Copyright License: © 2026 Sáez López JM, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

https://www.engineegroup.us

Check for updates

Abstract

This study examines the implementation of coding in a lifelong learning context through a progressive instructional approach that begins with simple activities using the visual block-based programming environment Scratch and gradually introduces the Unity game engine and the C# programming language through carefully sequenced tasks. A pre-experimental pre-test/post-test design was conducted with 54 participants enrolled in a university continuing education course. Two scales were administered to assess participants' perceptions regarding programming, game engines, and active learning. Descriptive statistics and inferential analysis were performed using the Wilcoxon signed-rank test. The findings reveal significant improvements in participants' perceptions of coding as a pedagogical strategy, particularly regarding creativity, educational innovation, motivation, active learning, and student engagement. The results also support the feasibility of a gradual progression from visual block-based programming to professional game development environments, highlighting the educational potential of combining Scratch and Unity in teacher training and lifelong learning programs.

Introduction

This intervention employs block-based programming (Scratch) in a straightforward manner that fosters successful pedagogy, subsequently allowing for a progressive transition to very simple activities within the Unity engine; this approach introduces a complex tool through an adapted, sequenced application.

A growing body of research and practice highlights the benefits of programming in elementary education [1-5]. Seymour Papert [6] pioneered this field, fostering procedural thinking in developing children through LOGO programming. Various authors [7] emphasise that programming is an extension of writing and, much like traditional writing, there are compelling reasons for everyone to learn to program.

The visual, block-based programming environment known as Scratch offers a significant advantage among languages designed to provide novices with their first introduction to computing; it allows users to "write" by dragging and dropping graphical blocks to assemble simple programs, which in turn enable them to create games, interactive stories, or simulations [8].

Evidence highlights the effectiveness and benefits of implementing programming activities that foster reflection [6,9]. Various studies have reported positive outcomes regarding attitudes toward informatics and computer science [10-12].

Unity offers a versatile and powerful development environment used by independent programmers [13]; consequently, students work with a pioneering, essential tool employed by industry professionals, thereby preparing them for future realities.

Several authors highlight the implementation of this engine in formal education [14], noting significant advantages to using such a comprehensive, professional-grade tool, alongside limitations stemming from the substantial time required to master it and the potential complexity of the tasks involved.

Citation: Sáez López JM, Sara RD. Programming with Scratch and Unity in Continuing Education: An Approach Featuring Simple Video Game Creation Activities. Trends Comput Sci Inf Technol. 2026;11(2):80-84. Available from: https://dx.doi.org/10.17352/tcsit.000113

Therefore, a pedagogical design and content selection that are as simplified as possible are recommended for elementary education.

Some institutions have reported success in game creation and development [15,16] and in motivating students as they create projects and assignments using computing [17,18].

This working method, in which students acquire technological skills and can create and manage their own projects, represents a constructivist framework that effectively activates prior knowledge, fosters metacognition through deeper information processing and enhanced understanding, and situates learning within a real-world context.

Objectives

The main objective of the study is to analyse the implementation and importance of visual block-based programming and the Unity engine in training activities.

The specific objectives are:

- To analyse capabilities related to programming and its learning process.
- To assess the feasibility of a block-based programming curriculum designed and sequenced to lead to basic proficiency with the Unity engine.
- To analyse the acquisition of basic computer programming concepts during training activities.

Methods

A pre-experimental research design with a pre-test/post-test approach was conducted to examine the educational impact of a progressive coding intervention based on visual block programming and game development. The study was implemented within a university lifelong learning course entitled *History of Video Games and their Implications in Education*, offered by the National Distance Education University (UNED, Spain).

Descriptive statistics were calculated for all questionnaire items. To determine differences between pre-test and post-test responses, the nonparametric Wilcoxon signed-rank test was applied because the study employed paired measurements from the same participants. Statistical significance was established at $\alpha = .05$.

Regarding the participants, the sample consisted of 54 participants enrolled in the course over two academic years. The group included 47.6% women and 52.4% men. All participants voluntarily completed the research instruments before and after the intervention.

The reliability of the research instrument was assessed using Cronbach's alpha coefficient to determine the internal consistency of the questionnaire. The analysis yielded a Cronbach's alpha value of 0.71, indicating an acceptable level of internal consistency according to commonly accepted psychometric standards. This result suggests that the questionnaire items demonstrate a satisfactory degree of homogeneity and are sufficiently correlated to measure the intended constructs consistently. Although the coefficient does not indicate excellent reliability, it exceeds the minimum threshold of 0.70 that is widely recommended for exploratory and educational research, supporting the suitability of the instrument for subsequent statistical analyses.

Therefore, the questionnaire can be considered a reliable tool for collecting data in the present study, providing sufficient confidence that the observed responses reflect the constructs under investigation rather than measurement error. Consequently, the instrument was deemed appropriate for further descriptive and inferential analyses.

Implementation

The intervention followed a progressive instructional sequence designed to facilitate the transition from introductory programming to the use of a professional game development environment. Participants first completed simple programming activities using Scratch, employing visual block-based programming to create interactive projects and educational games. Subsequently, they were introduced to the Unity game engine and basic C# programming through carefully scaffolded activities involving character movement, object interaction, and the development of a simplified version of the classic Pong video game.

The instructional design emphasised active learning, computational thinking, creativity, and problem-solving by engaging participants in the design and implementation of their own educational video games.

The intervention took place within the framework of the UNED course "History of Video Games and Their Implications for Education" (https://formacionpermanente.uned.es/tp_actividad/actividad/historia-de-los-videojuegos-y-sus-implicaciones-en-educacion), specifically through Task 3, which involved designing and developing a small educational video game using Scratch. This activity aimed to translate theoretical knowledge regarding video games and gamification into a practical creation experience, thereby fostering active learning and the development of digital competence. During the intervention, participants designed a video game with an educational purpose, first defining the learning objectives, theme, rules, and game mechanics. Subsequently, they programmed the project using Scratch blocks, incorporating characters, backdrops, events, variables, and scoring systems tailored to the chosen educational content.

The activity fostered the development of skills such as computational thinking [19], problem-solving, creativity, and digital resource planning [20]. It also provided insight into the potential of video games as learning tools by placing students in the role of educational experience designers rather than mere users. The course designates this task as an assessable activity accounting for one-third of the final grade, evaluating both the application of the material studied and the ability to create a functional educational resource.

Citation: Sáez López JM, Sara RD. Programming with Scratch and Unity in Continuing Education: An Approach Featuring Simple Video Game Creation Activities. Trends Comput Sci Inf Technol. 2026;11(2):80-84. Available from: https://dx.doi.org/10.17352/tcsit.000113

This training implementation proposes straightforward programming sessions using Scratch and Unity:

Walking movement from left to right. The project is accessible here: https://scratch.mit.edu/projects/604757836/ Walking movement from left to right. The project is published here: https://josemsaezlopez.itch.io/cat-walk.

A paddle that interacts in the style of "Arkanoid" (https://scratch.mit.edu/projects/616899990/) (Figure 1).

And finally, the most complex project: creating the game that launched the video game industry with Atari: *Pong*. In Scratch (https://scratch.mit.edu/projects/609271614/) and Unity (https://josemsaezlopez.itch.io/pong1) (Figure 2).

Results

Block-based programming and game engines in teacher training: Descriptive analysis and statistical inference using the Wilcoxon test

The analysis of Scale 1 “Block-based programming and game engines in education” reveals that items 1.1 and 1.3 show positive values (combining "agree" and "strongly agree"); thus, the majority believe that coding knowledge is essential for training future teachers and that interacting with game engines is feasible within teacher training.

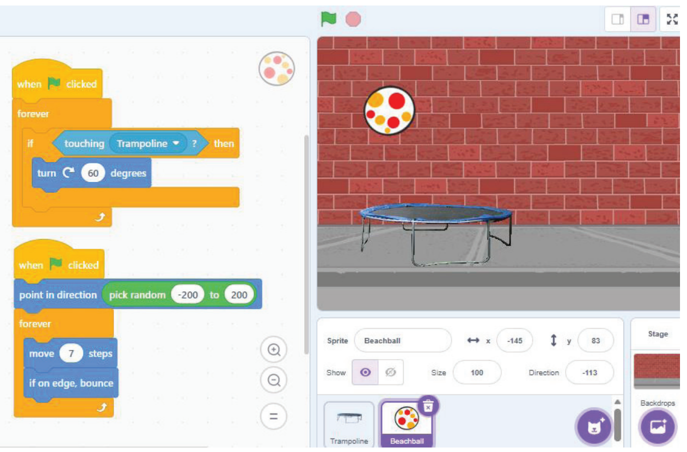

Figure 1: Paddle project in Scratch.

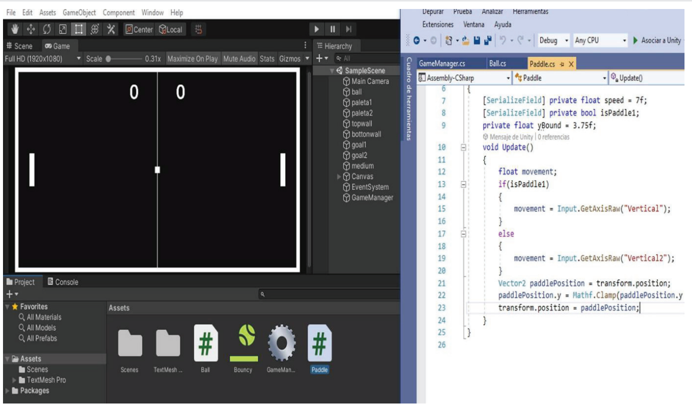

Figure 2: Pong Project in Scratch (https://scratch.mit.edu/projects/609271614/) and Unity (https://josemsaezlopez.itch.io/pong1)

Regarding statistical inference for this scale, the Wilcoxon test indicates a statistically significant improvement (at a significance level of α = 0.05) for items 1.1, 1.3, 1.4 and 1.5. Therefore, following the intervention, the subjects improved their attitudes toward programming knowledge in teacher training (1.1), gained a greater appreciation for the importance of sequencing with Scratch (1.3), innovation (1.4) and motivation (1.5) (Table 1).

Regarding Scale 1 (Block-based programming and game engines in training), the results show a very positive assessment by the participants. Specifically, 79.5% consider that the programming approach fosters creativity (item 1.1), 79.5% believe it promotes educational innovation processes (item 1.4), and 74.0% indicate that working with coding increases motivation in learning processes (item 1.5). Likewise, 62.9% consider a content sequence progressing from Scratch to Unity to be feasible (item 1.3), while 64.7% believe it is possible to work with complex, creative game engines like Unity (item 1.2).

Regarding statistical significance, statistically significant improvements (α = 0.05) are observed in the items related to creativity development (1.1), content sequencing from Scratch to Unity (1.3), educational innovation (1.4), and increased motivation through coding (1.5). Conversely, no statistically significant differences are observed for the item regarding the feasibility of working with complex game engines like Unity (1.2).

Regarding the "Active Learning" scale (Scale 2), participants express a highly favourable perception of using programming as a learning methodology. 85.0% believe that programming fosters active participation in the learning process (item 2.4), while 81.4% state that a coding-based approach makes the subject matter more interesting (item 2.3). Furthermore, 72.1% believe that significant content is learned through visual block-based programming (item 2.1), and a very similar percentage (73.9%) considers that substantial content is also acquired through creative game engines (item 2.2).

Statistically significant improvements (α = 0.05) are observed across all items comprising the "Active Learning" dimension (2.1, 2.2, 2.3, and 2.4) following the implementation of the program and the intervention; this highlights the positive impact of the training initiative on participants' perceptions of active learning.

Discussion and conclusions

The results reveal a highly favourable assessment of block-based programming as a tool for fostering creativity, motivation, and active learning. Regarding the dimension of block-based programming and game engines in education, agreement rates exceeding 70% across most items indicate that participants view programming as a methodology that goes beyond the acquisition of technical content, promoting transversal skills such as creativity and educational innovation. These findings align with Papert’s [6] constructionist philosophy, which posits that learning is more meaningful when students create products that hold personal significance.

Citation: Sáez López JM, Sara RD. Programming with Scratch and Unity in Continuing Education: An Approach Featuring Simple Video Game Creation Activities. Trends Comput Sci Inf Technol. 2026;11(2):80-84. Available from: https://dx.doi.org/10.17352/tcsit.000113

Table 1: Scales and items, descriptive analysis, Wilcoxon test. Values 1= Totally disagree, 2= Disagree, 3= Agree, 4= Totally agree.

Scales	Items	1	2	3	4	W.T.
1- Block-based programming and game engines in education.	1. The programming approach fosters the development of creativity.	3.7	16.6	38.8	40.7	.001
	2. It is possible to work with complex, creative game engines such as Unity.	9.2	25.9	44.4	20.3	.217
	3. Sequencing content from Scratch to Unity is feasible.	11.1	25.9	40.7	22.2	.004
	4. The programming approach facilitates educational innovation processes.	5.5	14.8	42.5	37.0	.001
	5. Working with coding increases motivation in learning processes.	3.7	22.2	40.7	33.3	.001
2- Active learning	1. A great deal of content is learned through visual block-based programming.	7.4	20.3	48.1	24.0	.001
	2. A great deal of content is learned using creative game engines.	3.7	22.2	46.2	27.7	.001
	3. The topic being studied becomes more engaging when approached through coding.	3.7	14.8	55.5	25.9	.001
	4. Programming fosters active participation.	5.5	9.2	49.9	35.1	.001

Similarly, the high value placed on creativity and motivation is consistent with the design of Scratch, an environment intended for students to learn by creating their own projects, thereby encouraging experimentation, problem-solving, and creative expression [7,8]. Existing literature also notes that visual programming environments eliminate many of the syntactic barriers found in traditional languages, allowing students to focus on designing solutions and developing computational thinking.

On the other hand, although participants consider the progression from Scratch to more complex development engines like Unity to be feasible, this aspect scores lower than the other dimensions; notably, the item regarding the direct use of Unity is the only one that does not show statistically significant differences. This result can be attributed to the greater technical complexity involved in using professional engines and the participants' limited prior experience with such tools. These findings align with research that highlights Scratch as an introductory environment while emphasising the need for appropriate methodological sequencing when transitioning to more complex languages or engines.

Regarding the "Active Learning" dimension, the results reveal broad consensus on programming's potential to boost student engagement and make the subject matter more interesting. These perceptions are consistent with research showing that creating video games and interactive projects fosters active, student-centred methodologies, thereby promoting autonomy, problem-solving skills, and meaningful learning [21]. Furthermore, a systematic review by Conde Melguizo et al. [22] identifies Scratch as one of the most widely used tools for developing digital competencies and computational thinking in educational settings, while also underscoring the need for further research employing more robust methodological designs.

Overall, the results of this study support the use of training initiatives based on video game design using Scratch, such as the one developed during the intervention, as they foster both the perception of programming's utility and the development of active methodologies oriented toward the creation, rather than merely the consumption, of digital resources.

The findings lead to the conclusion that the intervention based on creating a video game with Scratch was highly valued by the participants. The majority believe that programming fosters creativity, boosts motivation, and serves as a suitable tool for promoting educational innovation processes.

Furthermore, the intervention helped reinforce the perception that programming encourages active learning by increasing student participation and making the subject matter more engaging. The statistically significant differences observed across all items in this dimension underscore the effectiveness of the developed initiative.

On the other hand, while participants view a progression from Scratch to more advanced development engines like Unity as feasible, this sentiment is more tempered; this suggests the advisability of designing gradual training pathways that allow the fundamentals of visual programming to be consolidated first.

A limitation of the study is that the results are based on participant perceptions rather than objective measures of performance or the development of computational thinking. Consequently, future research could incorporate experimental designs, control groups, and longitudinal assessments to analyse the impact of such interventions on learning and the acquisition of digital competencies.

References

1. Fombona J, Sáez JM, Sánchez S. Artificial intelligence and robotics in education: advances, challenges, and future perspectives. Soc Sci Humanit Open. 2025;11. Available from: https://doi.org/10.1016/j.ssaho.2025.101533

2. Redondo-Duarte S, Pattier D, Neubauer A, Sáez López JM. Machine learning and educational robotics, an implementation in initial university teacher training and for practising teachers in primary education. Front Educ. 2026;11:1778718. Available from: https://doi.org/10.3389/feduc.2026.1778718

3. Sáez López JM, Redondo Duarte S, Neubauer Esteban A, Peña Garrido M. Teaching artificial intelligence through machine learning: an active learning approach in primary education classrooms. IJERI Int J Educ Res Innov. 2026;(25):1-14. Available from: https://doi.org/10.46661/ijeri.13119

4. Sáez-López JM, García-Jiménez AS, de-Lara-García-Cervigón S. Coding, robots, computational concepts, and machine learning using the microbit card

Citation: Sáez López JM, Sara RD. Programming with Scratch and Unity in Continuing Education: An Approach Featuring Simple Video Game Creation Activities. Trends Comput Sci Inf Technol. 2026;11(2):80-84. Available from: https://dx.doi.org/10.17352/tcsit.000113

and the Maqueen and Nezha kits: a study in initial teacher training. Comput Educ Open. 2026;10:100366. Available from: https://doi.org/10.1016/j.caeo.2026.100366

5. Sáez-López JM, Redondo-Duarte S, De Lara-García Cervigón S. Educational robotics with interactive and environmental projects using “micro: bit” and “Nezha” in nine primary schools. Eurasia J Math Sci Technol Educ. 2026;22(3). Available from: https://doi.org/10.29333/ejmste/18107

6. Papert S. Mindstorms: children, computers, and powerful ideas. New York (NY): Basic Books; 1980. Available from: https://worrydream.com/refs/Papert_1980_-_Mindstorms,_1st_ed.pdf

7. Resnick M, Maloney J, Hernández AM, Rusk N, Eastmond E, Brennan K, et al. Scratch: programming for all. Commun ACM. 2009;52(11). Available from: http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf.

8. Maloney J, Kafai Y, Resnick M, Rusk N. Programming by choice: urban youth learning programming with Scratch. In: Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education; 2008; Portland, OR. p. 367-371. Available from: https://doi.org/10.1145/1352135.1352260

9. Kolodner J, Camp P, Crismond D, Fasse B, Gray J, Holbrook J, et al. Problem-based learning meets case-based reasoning in the middle-school science classroom: putting Learning by Design into practice. J Learn Sci. 2003;12(4):495-547. Available from: https://doi.org/10.1207/S15327809JLS1204_2

10. Kwon DY, Kim HS, Shim JK, Lee WG. Algorithmic bricks: a tangible robot programming tool for elementary school students. IEEE Trans Educ. 2012;55(4):474-479. Available from: https://doi.org/10.1109/TE.2012.2190071

11. Lambert L, Guiffre H. Computer science outreach in an elementary school. J Comput Sci Coll. 2009;24(3):118-124.

12. Lin JMC, Yen LY, Yang MC, Chen CF. Teaching computer programming in elementary schools: a pilot study. In: National Educational Computing Conference; 2005 Jun.

13. Dickson PE. Using Unity to teach game development: when you’ve never written a game. In: Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE ’15). New York (NY): ACM; 2015. p. 75-80. Available from: https://doi.org/10.1145/2729094.2742591

14. Comber O, Motschnig R, Mayer H, Haselberger D. Engaging students in computer science education through game development with Unity. In: 2019 IEEE Global Engineering Education Conference (EDUCON). IEEE; 2019. p. 199-205. Available from: https://doi.org/10.1109/EDUCON.2019.8725135

15. Kessler R, van Langeveld M, Altizer R. Entertainment arts and engineering (or how to fast-track a new interdisciplinary program). In: Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE ’09); 2009 Mar 4-7; Chattanooga, TN. New York (NY): ACM; 2009. p. 539-543. Available from: https://doi.org/10.1145/1508865.1509049

16. Linhoff J, Settle A. Teaching game programming using XNA. In: Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE ’08); 2008 Jun 30-Jul 2; Madrid, Spain. New York (NY): ACM; 2008. p. 250-254.

17. Israel M, Pearson JN, Tapia T, Wherfel QM, Reese G. Supporting all learners in school-wide computational thinking: a cross-case qualitative analysis. Comput Educ. 2015;82:263-279.

18. Sáez-López JM, Román M, Vázquez E. Visual programming languages integrated across the curriculum in elementary school: a two-year case study using Scratch in five schools. Comput Educ. 2016;97:129-141. Available from: https://doi.org/10.1016/j.compedu.2016.03.003

19. Wing JM. Computational thinking. Commun ACM. 2006;49(3):33-35. Available from: https://doi.org/:10.1145/1118178.1118215

20. Gómez Jiménez Ó, Sáez López JM, Rodríguez Torres J. Autopercepción de la competencia digital en estudiantes de Educación Primaria. Educ Knowl Soc. 2026;27. Available from: https://doi.org/10.14201/eks.31923

21. Sáez-López JM, Cózar R. Programación visual por bloques en Educación Primaria: aprendiendo y creando contenidos en Ciencias Sociales. Rev Complut Educ. 2017;28(2):409-426. Available from: https://doi.org/10.5209/rev_RCED.2017.v28.n2.49381

22. Conde Melguizo R, Vega Barbas M, García Vázquez C. Analizando el auge de Scratch para la enseñanza de la programación: revisión del conocimiento científico publicado en España. Tarbiya Rev Investig Innov Educ. 2020;(48):7-32. Available from: https://doi.org/10.15366/tarbiya2020.48.001

23. Brennan K, Resnick M. Using artefact-based interviews to study the development of computational thinking in interactive media design. Paper presented at: American Educational Research Association Meeting; 2012; Vancouver, BC, Canada. Available from: http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf.

24. Bruckman A. Learning in online communities. In: Sawyer K, editor. Cambridge handbook of the learning sciences. New York (NY): Cambridge University Press; 2006. p. 461-472.

25. Clark J, Rogers MP, Spradling C, Pais J. What, no canoes? Lessons learned while hosting a Scratch summer camp. J Comput Sci Coll. 2013;28:204-210.

26. Cózar-Gutiérrez R, Sáez-López JM. Game-based learning and gamification in initial teacher training in the social sciences: an experiment with MinecraftEdu. Int J Educ Technol High Educ. 2016;13:2. Available from: https://doi.org/10.1186/s41239-016-0003-4

27. Kafai Y. Children as designers, testers and evaluators of educational software. In: Druin A, editor. The design of children’s technology. San Francisco (CA): Morgan Kaufmann Publishers; 1998. p. 123-145. Available from: https://dl.acm.org/doi/10.5555/303430.303438

28. Sáez-López JM, Del Olmo J, Gonzalez-Calero JA, Cozar R. Exploring the effect of training in visual block programming for preservice teachers. Multimodal Technol Interact. 2020;4(3):65. Available from: https://doi.org/10.3390/mti4030065

Discover a bigger Impact and Visibility of your article publication with Peertechz Publications

Highlights

- Signatory publisher of ORCID
- Signatory Publisher of DORA (San Francisco Declaration on Research Assessment)
- Articles archived in worlds’ renowned service providers such as Portico, CNKI, AGRIS, TDNet, Base (Bielefeld University Library), CrossRef, Scilit, J-Gate etc.
- Journals indexed in ICMJE, SHERPA/ROMEO, Google Scholar etc.
- OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting)
- Dedicated Editorial Board for every journal
- Accurate and rapid peer-review process
- Increased citations of published articles through promotions
- Reduced timeline for article publication

Submit your articles and experience a new surge in publication services
https://www.peertechzpublications.org/submission

Peertechz journals wishes everlasting success in your every endeavours.

Citation: Sáez López JM, Sara RD. Programming with Scratch and Unity in Continuing Education: An Approach Featuring Simple Video Game Creation Activities. Trends Comput Sci Inf Technol. 2026;11(2):80-84. Available from: https://dx.doi.org/10.17352/tcsit.000113