

ROBLOX STUDIO INTEGRATION WITH 3D MODELING SOFTWARE

Dragan Rastovac¹ Božo Ilić² Ana Bašić³

Abstract: Technology is rapidly advancing, with particular emphasis on the innovative development of information and communication technology (ICT), which significantly impacts all layers of society (both adults and children). This paper presents a creative example of developing a video game in Roblox Studio, incorporating programming elements and integrating it with the 3D modeling program Blender. The analysis will be descriptive and supported by practical examples. Roblox Studio is a program used not only for playing video games but also for creating them on the Roblox platform. In this way, children express their creativity and imagination in developing their own games while also playing games created by other platform users, exchanging experiences in the process.

Key words: computer science, primary school, programming, learning, Roblox Studio.

1. INTRODUCTION

The development of technology, particularly information and communication technology (ICT), has influenced society as a whole. A large number of different manufacturers have contributed to making ICT accessible to all social groups. As a result, children have become increasingly "enchanted" by digital devices such as tablets, phones, and computers. Their development and free time are increasingly directed toward applications offered by digital devices, while they spend less time socializing and playing with their peers.

In addition to entertainment, children also use ICT for learning, both at home and in school. Access to these devices provides them with a vast amount of information [1]-[4].

According to the authors [5], based on [6], [7], modern ICT introduces new and more reliable ways of distributing, presenting, and storing "communication." It also defines applications and resources, changing standard communication practices both at school and at home. All of this has, in a way, prompted the Ministry of Education in Serbia to adapt to the ICT era by integrating Informatics as a mandatory subject in the official elementary school curriculum.

2. COMPUTER SCIENCE AND INFORMATICS IN ELEMENTARY SCHOOL

The structured implementation of informatics in the educational process is based on the K-12 Computer Science (CS) curriculum. This has become a new discipline that facilitates lesson planning for teachers by defining key areas that should be covered in their teaching units. Additionally, it helps educators prepare for various topics included in the CS curriculum.

The first countries to implement computer science in schools were the United States, the United Kingdom, and Australia [8]. A detailed structure of the CS K-12 curriculum can be found in [9].

Authors of the study [10], based on the "Report of the Joint Informatics Europe & ACM Europe Working Group," argue in favor of introducing informatics in elementary schools. They

¹ Doctor of Electrical Engineering and Computer Science, Higher Education Technical School of Professional Studies in Novi Sad, rastovac@vtsns.edu.rs

² Doctor of Technical Sciences, Higher Education Technical School of Professional Studies in Novi Sad, bozoilic66@gmail.com

³ doctor ITS Belgrade, ana.basic@its.edu.rs

particularly emphasize the importance of computational thinking, which can be a key factor in developing future experts in computer science. The significance of computational thinking was highlighted by Wing in his study [11], where he argues that everyone should possess computational thinking skills, as it greatly influences analytical abilities (especially in children), activates mental tools, and enhances understanding of human behavior.

In Serbia, since the 2017/18 school year, informatics has been taught as a mandatory subject (1 hour per week) for students from the fifth to the eighth grade under the name "Informatics and Computing." Additionally, starting from the 2020/21 school year, the subject "Digital World" (1 hour per week) became a mandatory subject for students from the first to the fourth grade [12], [13].

3. PROGRAMMING IN ELEMENTARY SCHOOL

Programming is considered highly attractive and offers promising career prospects. The demand for programmers in the job market is constantly growing, prompting many young people to attend programming courses to reskill and transition from their current professions into programming careers [14].

Many countries have integrated Scratch into their elementary school curricula to teach children programming. Scratch provides a graphical environment that simplifies coding by using block-based programming. Other similar tools, such as Snap! [15] and Alice [16], are used less frequently. Block-based programming involves dragging and dropping command blocks to create code, as illustrated in Figure 1. This method is commonly referred to as visual or graphical programming, making it easier for children to understand programming concepts without dealing with complex syntax.

Studies analyzing the use of Scratch in education can be found in [18]-[21].



Figure 1 – Example of Block Code in Scratch (adapted from [22])

When it comes to text-based programming languages such as Python and Java, they are generally avoided in elementary school (K-12 education) as they are considered too complex for beginners. These programming languages are typically seen as professional tools meant for individuals with prior programming experience [23].

However, some research studies suggest that introducing text-based programming at an early age can actually enhance students' motivation, improve their understanding of algorithms, and strengthen logical reasoning skills [24], [25].

The authors of study [26], based on conclusions from [27]-[29], emphasize the importance of computational thinking in elementary school students. It is considered a powerful tool for enhancing creative thinking and problem-solving skills in the ICT era.

Educational experts recommend that computational thinking be incorporated into the curriculum for younger elementary school students. This approach helps students develop skills to better analyze problems, find solutions, and navigate the digital world and future technologies effectively.

4. INTEGRATION OF ROBLOX STUDIO WITH BLENDER

Roblox Studio was designed by Roblox Corporation and serves as a social gaming platform. According to [30] and based on [31], Roblox is a large virtual platform for multiplayer gaming, offering interactive virtual worlds where users can create and share their own video games with the community. This feature makes Roblox a unique user-generated content platform, allowing developers and players to engage in a shared creative ecosystem. On the other hand, Blender is a 3D modeling and animation software that supports game creation, animation, rendering, and sculpting. It is a cross-platform application, compatible with multiple operating systems, including Linux and Windows [32].

By integrating Roblox Studio with Blender, developers and students can enhance their game assets, creating customized 3D models, characters, and environments for more immersive experiences.

A practical example will demonstrate the integration of Blender with Roblox Studio, showcasing how specific objects and their components can be improved using Blender. This process enhances the physical and visual quality of the objects compared to the default assets available in Roblox Studio.

By leveraging Blender's advanced modeling tools, developers can create more detailed and realistic assets, which significantly improves the overall game aesthetics.

Figure 2 presents a comparative view of a "hand" model: On the left side of Figure 2 is the default hand model provided in Roblox Studio templates. On the right side is a hand model created in Blender, demonstrating improved geometry, smoother surfaces, and greater detail.

This comparison highlights how Blender can enhance game assets beyond the standard options available in Roblox Studio, allowing developers to personalize and optimize their creations.

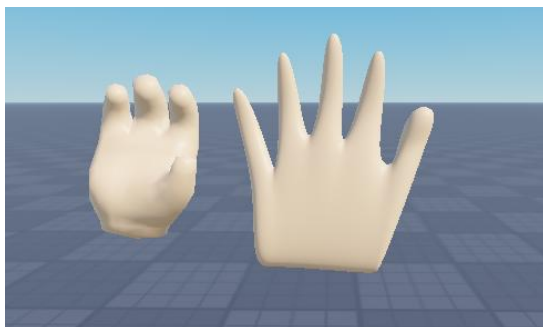


Figure 2 – Comparative View of a "Hand"

Figure 3 illustrates the visual design and key objects that are part of the created video game. The game features an interactive environment where the player must manage resources to keep their pet happy and well-fed. Object 1 (Player Character): Controlled by the player, takes the form of a human-like figure, responsible for collecting money to buy food for the pet. Object 2 (Pet – Cube-Shaped): Represents a pet, visually designed as a cube, displays three indicators above it: age, hunger level, happiness level. Money Collection Mechanic: The player clicks on Object 2 (pet) to generate money, Money appears on the ground, and the player has limited time to collect it. Object 3 (Computer on a Table): Serves as the in-game store where players can purchase food for their pet. The cost of food is based on the amount of collected money. Figure 4 presents the food bowl, modeled using Blender. This object is used to feed the pet, impacting its happiness and hunger levels. By integrating Roblox Studio and Blender, this game demonstrates how custom 3D assets enhance the overall experience, making gameplay more immersive and visually engaging.



Figure 3 – The Appearance of the Created 3D Game in Roblox.



Figure 4 – The Appearance of the Created Food Bowl in Blender

5. CONCLUSION

It goes without saying that the development of a school curriculum is directly tied to national education policies. Additionally, the successful implementation of educational activities relies on well-trained teachers who can effectively deliver content. Certain countries, particularly Finland and Norway, consider computational thinking a fundamental skill that should be integrated into all subjects of elementary education, rather than being limited to informatics alone [33].

This study has demonstrated an integrated approach by combining Roblox Studio with the Blender 3D modeling software. The practical example showcases the versatility of Roblox Studio, enabling game creators to enhance object visualizations if they find the built-in assets insufficient. This approach highlights the flexibility and creativity of game developers, who can customize and refine their game assets to create a more engaging experience. By doing so, developers can attract more players, making their game more visually appealing and immersive.

6. REFERENCES

- [1] OECD. Students: *Computers and Learning: Making the Connection*. Paris: PISA, OECD Publishing; 2015. <https://doi.org/10.1787/9789264239555-en>.
- [2] Becker, B.: *Educational ICT use outside school in the European Union: disparities by social origin, immigrant background, and gender*. J Children and Media 2020.
- [3] Gracia P.; Garcia-Roman J.; Oinas T.; Anttila T.: *Child and adolescent time use: A cross-national study*. J Marriage and Family 2020; 82(4):1304–25.
- [4] Borgonovi, F.; Pokropek, M.: *The evolution of the association between ICT use and reading achievement in 28 countries*. Computers and Education Open, 2, 100047, 2021.
- [5] Heath, D.; Maghrabi, R.; Carr, N.: *Implications of information and communication technologies (ICT) for school-home communication*. Journal of Information Technology Education: Research, 14, 2015.
- [6] Wright, D. K.: *The Magic Communication Machine: Examining the Internet's Impact on Public Relations, Journalism, and the Public*. Alabama, 2001.
- [7] Pavlik, J. V.: *Mapping the consequences of technology on public relations*. Institute for Public Relations, 2007, pp. 1-20.
- [8] Vivian, R.; Quille, K.; McGill, M. M.; Falkner, K.; Sentance, S.; Barksdale, S.; ... & Maiorana, F.: *An international pilot study of k-12 teachers' computer science self-esteem*. In Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, 2020, pp. 117-123.
- [9] Frost, D.; Verno, A.; Burkhart, D.; Hutton M.; North, K.: *A Model Curriculum for K-12 Computer Science: Level I Objectives and Outlines*, CSTA Curriculum Committee, 2009.
- [10] Sabitzer, B.; Antonitsch, P. K.; Pasterk, S.: *Informatics concepts for primary education: preparing children for computational thinking*. In Proceedings of the 9th Workshop in Primary and Secondary Computing Education, 2014, pp. 108-111.
- [11] Wing, J. M.: *Computational thinking*. Communications of the ACM, 49(3) , 2006, pp. 33-35.
- [12] <https://www.srbija.gov.rs/vest/290575/informatika-od-naredne-skolske-godine-obavezan-predmet.php>, November 2024.
- [13] <https://prosveta.gov.rs/vesti/digitalni-svet-za-buduce-prvake/>, November 2024.
- [14] Robins, A.; Rountree, J.; Rountree, N.: *Learning and teaching programming: A review and discussion*. Computer science education, 13(2), 2003, pp. 137-172.

- [15] Harvey, B.; Garcia, D.; Paley, J.; Segars, L.: *Snap!(build your own blocks)*. In Proceedings of the 45th ACM technical symposium on Computer science education. ACM, 2014, pp. 749–749.
- [16] Cooper, S.; Dann, W.; Pausch, R.: *Teaching Objects-first in Introductory Computer Science*. SIGCSE Bull. 35, 1, 2003, pp. 191–195.
- [17] Fagerlund, J.; Häkkinen, P.; Vesisenaho, M.; Viiri, J.: *Computational thinking in programming with Scratch in primary schools: A systematic review*. Computer Applications in Engineering Education, 29(1), 2021, pp.12-28.
- [18] Rastovac, D.; Ilić, B.: (2024). *Text-based programming for sixth-grade elementary school students. (Tekstualno programiranje za decu šestog razreda osnovne škole) - ROBLOX STUDIO*. Proceedings of the 40th Conference with International Participation of Maintenance Professionals of Serbia and the 3rd Advanced Technologies in the Function of the Economy, May 28-30, 2024, Vrnjačka Banja, 2024, pp. 311-316 (Zbornik radova 40. konferencije sa međunarodnim učešćem Održavalaca Srbije i 3. Napredne tehnologije u funkciji privrede, 28-30. maj 2024, Vrnjačka Banja, 2024, pp. 311-316.
- [19] Cheng, G.: *Exploring factors influencing the acceptance of visual programming environment among boys and girls in primary schools*. Computers in Human Behavior, 92, 2019, pp. 361-372.
- [20] Kong, S. C.; Wang, Y. Q.: *Assessing programming concepts in the visual block-based programming course for primary school students*. In Proceedings of the European Conference on e-Learning, ECEL, 2019, pp. 294-302.
- [21] Sáez-López, J. M.; Román-González, M.; Vázquez-Cano, E.: *Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools*. Computers & Education, 97, 2016, pp. 129-141.
- [22] <https://www.codeyoung.com/blog/how-to-make-a-game-on-scratch>, novembar 2024.
- [23] Humble, N.: *Using textual programming tools to develop computational thinking skills in K-12 education*. In EDULEARN20 Proceedings, 2020, IATED, pp. 7188-7195.
- [24] Nouri, J.; Zhang, L.; Mannila, L.; Norén, E.: *Development of computational thinking, digital competence and 21st century skills when learning programming in K-9*. Education Inquiry, 11(1), 2020, pp. 1-17.
- [25] Tsukamoto, H.; Takemura, Y.; Nagumo, H.; Ikeda, I.; Monden, A.; Matsumoto, K. I.: *Programming education for primary school children using a textual programming language*. In 2015 IEEE Frontiers in Education Conference (FIE), IEEE, 2015, pp. 1-7.
- [26] Zhao, L.; Liu, X.; Wang, C.; Su, Y. S.: *Effect of different mind mapping approaches on primary school students' computational thinking skills during visual programming learning*. Computers & Education, 181, 104445, 2022.
- [27] Kong, S. C.: *Components and methods of evaluating computational thinking for fostering creative problem-solvers in senior primary school education*. Computational thinking education, 2019, pp. 119-141.
- [28] Lindberg, R. S.; Laine, T. H.; Haaranen, L.: *Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games*. British journal of educational technology, 50(4), 2019, pp. 1979-1995.
- [29] Lai, Y. H.; Chen, S. Y.; Lai, C. F.; Chang, Y. C.; Su, Y. S.: *Study on enhancing AIoT computational thinking skills by plot image-based VR*. Interactive Learning Environments, 29(3), 2021, pp. 482-495.
- [30] Tinterri, A.; Guerriero, M. A.; Annoscia, S., & Dipace, A.: *Constructionism and game-making for learning in the age of Roblox*, An analysis of current evidence and future perspectives, GIORNALE ITALIANO DI EDUCAZIONE ALLA SALUTE, SPORT E DIDATTICA INCLUSIVA, 7(2), 2023, pp. 1-16.
- [31] Dwivedi, Y. K., Hughes, L., Baabdullah, A. M., Ribeiro-Navarrete, S., Giannakis, M., Al-Debei, M. M., ... & Wamba, S. F.: *Metaverse beyond the hype: Multidisciplinary perspectives on emerging*

challenges, opportunities, and agenda for research, practice and policy, *International Journal of Information Management*, 66, 2022, 102542.

[32] <https://www.blender.org/about/>, October 2024.

[33] Dagienė, V.; Hromkovič, J.; Lacher, R.: *Designing informatics curriculum for K-12 education: From Concepts to Implementations*. *Informatics in Education*, 20(3), 2021, pp. 333-360.