Game based learning approach for teaching computational thinking and programming concepts for vocational high school students

Tatu Toikkanen

[a] *University of Jyväskylä,*

Abstract

Abstract will be here.

Keywords: Computational thinking, Game based learning, Gamification, Scratch, Teaching methods

1. Introduction

Teaching computational thinking to students presents challenges, including integrating computational thinking concepts into curricula and developing engaging teaching methodologies. Additionally, equipping educators with the requisite skills and knowledge is paramount. (Angeli and Giannakos, 2020)

This teaching experiment utilizes a game-based learning approach to address these challenges among first-year vocational high school students, as part of the intermediate studies in the University of Jyväskylä's Master's Degree Program in Education Technology.

2. Context

In our digital era, the significance of computational thinking and programming abilities across diverse sectors cannot be overstated. The task of imparting these essential skills is challenged by the need to develop curricula that effectively incorporate computational thinking, to enhance the preparedness and expertise of educators in these domains, and to ensure the availability of sufficient tools and resources that support learning in varied educational settings. (Angeli and Giannakos, 2020)

As educational paradigms evolve, there's a growing appreciation for the role of game-based learning in boosting student engagement, motivation, and outcomes. (Greipl et al., 2020) Furthermore, gamification has shown promise in not only elevating motivation but also in streamlining the learning process. (Liu et al., 2020)

Computational thinking is a crucial skill set for the analysis and resolution of complex issues through computational methods, embodying algorithmic thinking, problem decomposition, pattern recognition, and abstraction. (Wing, 2006)

Vocational high schools, catering to students with diverse academic and career ambitions, often struggle to engage their students in abstract subjects such as programming using traditional pedagogical methods. (Freeman et al., 2014)

Game-based learning offers a viable solution to these educational hurdles, utilizing gaming to support instruction and learning, thereby fostering engagement through interactive and immersive experiences. (Prensky, 2009)

This teaching experiment evaluates the effectiveness of a game-based learning approach in teaching computational thinking and programming concepts to vocational high school students. It seeks to overcome the educational challenges by leveraging engaging and motivational tools and resources, enhancing the overall learning experience.

3. Method

This teaching experiment was conducted as part of the University of Jyväskylä's Master's Degree Program in Education Technology. The core objective of the study was to evaluate the effectiveness of a game-based learning approach for teaching computational thinking and programming concepts to vocational high school students. This was achieved by utilizing Scratch as the primary educational tool within a classroom setting, aiming to measure student engagement, learning outcomes, and the development of computational thinking skills through the creation and execution of games and interactive projects.

The participant group consisted of first-year students from an Information and Communications Technology (ICT) program at a vocational high school. This group was notably diverse, ranging in age and educational background, including those who had recently completed grades 7 to 9 and others who were undergoing retraining for a new career path. Such diversity represented a wide array of abilities and interests among the students. Despite the significant age range of 16 to 38 years within the study population, the majority of participants were 16 or 17 years old, indicating a mainly younger group.

3.1. Curriculum structure

The curriculum, designed to span four hours across four distinct sessions, focuses on introducing students to the foundational elements of programming, computational thinking, and algorithmic reasoning. It leverages Scratch, a visual programming language created by the MIT Media Lab (Resnick et al., 2009), to immerse students in the learning of programming concepts through the engaging approach of gamification. The choice of Scratch is deliberate; its user-friendly interface and emphasis on fostering creativity render it suitable for familiarizing students with the basics of programming. (Maloney et al., 2010) This structured curriculum aims to enhance students' grasp of programming constructs, algorithmic thinking, and the crafting of simple games and animations.

To facilitate this curriculum, access to a computer lab equipped with internet connectivity is essential, ensuring each student can utilize the Scratch online platform (scratch.mit.edu). Teachers are tasked with providing comprehensive instructional materials, which include an introductory overview of Scratch and a selection of project examples, thereby supporting a hands-on learning experience that encourages experimentation and exploration within the realm of programming.

3.1.1. Hour 1: Introduction to Programming and Scratch

Objective: Introduce Scratch's interface and basic programming concepts.
Activities: Overview of programming concepts and a guided exploration of Scratch, followed by a simple project creation (Resnick et al., 2009).

3.1.2. Hour 2: Fundamentals of Algorithmic Thinking

Objective: Develop basic algorithmic thinking skills.
Activities: Construction of algorithms using Scratch blocks, and an exercise to create a small game or animation applying loops and conditions (Maloney et al., 2010).

3.1.3. Hour 3: Designing Games and Animations

Objective: Understand the process of designing and implementing games and animations in Scratch.
Activities: Individual project focusing on game mechanics and character control.

3.1.4. Hour 4: Project Presentation and Conclusion

Objective: Present projects and reflect on the learning process.
Activities: Presentation of student projects, discussion of learning experiences, and course wrap-up.

3.2. Evaluation

In this teaching experiment, the evaluation of teaching methodologies is primarily informed by student-provided data, leveraging a methodical framework that prioritizes firsthand feedback regarding the effectiveness and influence of the applied instructional strategies.

To accurately measure the effectiveness of the teaching methods, we use a combination of quantitative data collection technique. Students are asked to complete evaluation forms that specifically address the instructional methods used in the experiment. This form is designed to measure various aspects of the teaching methods, such as their clarity, engagement level, and appropriateness for the content delivered, without directly assessing the personal or professional attributes of the instructor.

How effectively were Scratch projects integrated into learning activities for understanding programming concepts? Very Ineffectively 1 - 5 Very Effectively
Rate the quality and usefulness of Scratch programming resources provided for the course. Very Poor 1 - 5 Excellent
How much did the teaching methods encourage experimentation and creativity with Scratch? Not at All 1 - 5 Always
Did the teaching methods effectively demonstrate the real-world application of programming and computational thinking skills learned through Scratch? Not at All 1 - 5 Very Effectively
How constructive was the feedback provided on Scratch projects for improving your programming skills? Not Constructive 1 - 5 Very Constructive
Considering everything, how effective do you find the teaching methods for this Scratch programming course? Very Ineffective 1 - 5 Very Effective

Table 1: Game based learning approach for teaching computational thinking and programming concepts - Translated evaluation form

Consistent with the principles of evaluation research, our approach is to execute a systematic data collection process designed to yield insights into the efficiency of game-based learning techniques in a vocational high school context, specifically focusing on the computational thinking and programming fundamentals. (Powell, 2006)

The collection of evaluation data was facilitated through an online questionnaire encompassing six items. Participants evaluated each aspect on a scale from one to five, indicating their varying perceptions of the educational experience. A score of five on the scale indicates a highly positive and exceptional learning outcome. However, it's crucial to note that the significance given to each score varied across the questions. All participants were required to answer every question on the questionnaire. A detailed summary of the questions and their possible responses is provided, facilitating a thorough understanding of the evaluation metrics.

4. Results

5. Discussion

6. Summary and conclusions

Acknowledgements

I extend my gratitude to Pohjoisen-Keski Suomen ammattiopisto for facilitating this experiment and offer special thanks to all the students who participated, making this study possible.

References

Angeli, C., Giannakos, M., 2020. Computational thinking education: Issues and challenges.

Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H., Wenderoth, M.P., 2014. Active learning increases student performance in science, engineering, and mathematics. Proceedings of the national academy of sciences 111, 8410–8415.

Greipl, S., Moeller, K., Ninaus, M., 2020. Potential and limits of game-based learning. International Journal of Technology Enhanced Learning 12, 363–389.

Liu, Z.Y., Shaikh, Z.A., Gazizova, F., 2020. Using the concept of game-based learning in education. International Journal of Emerging Technologies in Learning .

Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E., 2010. The scratch programming language and environment. ACM Transactions on Computing Education (TOCE) 10, 1–15.

Powell, R.R., 2006. Evaluation research: An overview. Library trends 55, 102–120.

Prensky, M., 2009. H. sapiens digital: From digital immigrants and digital natives to digital wisdom. Innovate: journal of online education 5.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al., 2009. Scratch: programming for all. Communications of the ACM 52, 60–67.

Wing, J.M., 2006. Computational thinking. Communications of the ACM 49, 33–35.