

Refurbishing the Educational Escape Room for Programming: Lowering the Threshold and Raising the Ceiling

Niklas Humble and Peter Mozelius
Mid Sweden University, Östersund, Sweden

niklas.humble@miun.se

peter.mozelius@miun.se

Abstract: Programming education at university level has often been identified as problematic learning. At the same time, prognosis for future labour market is increased need for professionals with programming and related skills. To meet demands of future society, K-12 schools around the world have integrated programming in the curriculum. However, research show challenges for integrating programming in K-12 education. Challenges include students' and teachers' struggle with learning and teaching programming, lack of time to properly incorporate programming in existing teaching and learning, and insufficient support for teachers. This study addresses these challenges by designing, developing, and evaluating an educational game on programming that combines the idea of game-based learning with digital escape rooms. In the game, the player develops knowledge about programming concepts to escape 10 rooms. The study's main research question was: What are K-12 students' perceptions of the game and what do they considered to be important design factors for a digital escape room game on computer programming? A design science approach was used for designing, developing, and evaluating a web-based escape room game on programming. This was conducted in a five-step process: 1) Explicating the problem, 2) Defining the requirements, 3) Designing and developing the artefact, 4) Demonstrating the artefact, and 5) Evaluating the artefact. The game was tested and evaluated by 32 K-12 students with a questionnaire during the autumn semester of 2021 and spring semester of 2022. Collected data were then analysed and grouped into categories to answer the study's aim and research question. Findings of the study show several suggestions for further development and important design factors to consider when developing a digital escape room game. The next steps of research are to combine these findings with evaluations from teachers, and to incorporate this in an updated version of the game.

Keywords: Escape rooms, Game development, Game-based learning, Programming education, K-12 education

1. Introduction

Computer programming has been addressed in previous research as problematic learning, regarding student drop-out and satisfaction in higher education (Pereira et al., 2019; Marcolino & Barbosa, 2017; Gomes & Mendes, 2007). At the same time, research highlights the need of professionals with STEM (Science, Technology, Engineering and Mathematics) and 21st century skills on future labour market, in which programming is an important part (Tikva & Tambouris, 2021; Admiraal et al., 2019). To meet the demands of future labour market and society, a global trend has been to integrate programming in kindergarten to grade 12 (K-12) education (Tikva & Tambouris, 2021; Szabo et al., 2019). In Sweden, the integration of programming in K-12 curriculum was decided and outlined by the Swedish government in 2017 (Nouri et al., 2020; Heintz et al., 2017).

However, research show challenges for teaching and integrating programming also at K-12 level. In the UK, computing education has been described as "patchy and fragile" with lack of teacher support (The Royal Society, 2017). Research have further described a lack of time for the integration and challenges for both teachers and students to learn programming (Allison, 2021; Humble, Mozelius & Sällvin, 2020). To address the challenges of integrating programming in K-12 education, this study describes the process of designing, developing, and evaluating an educational game on programming. The game design combines the idea of game-based learning (Hussein et al., 2019) with that of digital escape rooms (Veldkamp et al., 2020) to foster student understanding of basic concepts within computer programming.

With the design science idea of iterative development this study could be seen as the second iteration of the game development presented by Humble, Mozelius & Sällvin (2021). Striving for a more inclusive game design, authors have had the ambition of lowering the threshold and raising the ceiling of educational escape rooms on programming. That is, getting more students interested in programming through educational escape rooms (lowering the threshold) while still facilitate learning and engagement in programming (raising the ceiling). The aim of the study was to design, develop and evaluate a digital escape room game on computer programming for K-12 students, with the main research question: What are K-12 students' perceptions of the game and what do they considered to be important design factors for a digital escape room game on computer programming?

2. Extended background

Computer programming has been identified as a subject that many students find challenging and problematic to learn (Chetty & Barlow-Jones, 2012; Hawi, 2014; Malliarakis, Shabalina & Mozelius, 2021). An ongoing reform that will facilitate programming education at university level is the introduction of programming in K-12 education. At the same time, challenges associated with introductory programming will be shifted to the younger audience, with an even more urgent need for pedagogy shift and curriculum development (Lédeczi et al., 2021). This paper suggests combining the old tradition of game-based learning with the current trend of educational escape rooms.

2.1 Game-based learning

The tradition of using games and play in educational contexts is far older than using computers. Various types of games have for thousands of years been used for training in areas such as mathematics, logics, and combinatorics (Vankúš, 2005; Hellerstedt & Mozelius, 2019). Areas that all are useful in the process of developing fundamental programming skills and for the general understanding of computer science. In computer science, the construction and analysis of games have resulted in creation of valuable knowledge. Two prominent computer scientists that have been pioneers in this field are the chess aficionados and creative game developers Alan Turing (1953) and Ken Thompson (1986). In programming, educational games have been involved in teaching and learning activities both for learning by creating games (Seralidou & Douligieris, 2021), and for learning to program by playing games (Miljanovic & Bradbury, 2018). This could also be carried out for the younger audiences (Gomes, Falcão & Tedesco, 2018), with the use of open educational games (Silva & Silveira, 2020). A related concept is that of flow and the flow channel, which is the state between anxiety and boredom where the player experience peaked enjoyment and focus (Csikszentmihalyi, 1975; Neal, 2012).

2.2 Educational escape rooms

The much younger trend of Escape rooms started out as leisure gaming, and often with the purpose of team building. With the aim of keeping the fun, the escape room concept has also been reused for different educational purposes such as chemistry (Yayon et al., 2019) and language learning (Urbieta & Peñalver, 2019). What inspired this study was that escape rooms have showed promising learning support in the domains of science, technology, engineering, and mathematics (STEM), and the computer science domain (Borrego et al., 2017). The STEM disciplines have acted as pioneers in the construction and use of educational escape rooms, with a frequent use of escape rooms in teaching and learning activities (Veldkamp et al., 2020). Finally, and even more inspiring, the escape room concept has also been tested with promising results in programming courses (López-Pernas et al, 2019), both for face-to-face activities and in distance mode (López-Pernas et al, 2021).

3. Method

Hevner et al. (2004) describe design science as “a problem-solving paradigm” that seeks to innovate ideas, practices, and technical artefacts. A common goal of design science is to produce design knowledge, that is, knowledge that support problem-solving in the design of solutions (Engström et al., 2020). A design science approach, inspired by Johannesson and Perjons (2014), was used to structure this study in five steps:

1. Explicate the problem
2. Define requirements
3. Design and develop artefact
4. Demonstrate artefact
5. Evaluate artefact

The first step, to explicate the problem, is addressed in the sections *1. Introduction* and *2. Extended background* of this study. The problem was also identified through discussions in professional development courses on fundamental programming for K-12 teachers, where the authors worked as course facilitators (Humble, Mozelius & Sällvin, 2021). In the second step, the requirements for the learning outcomes of the game were designed to closely align with the syllabus framework for the previously mentioned teacher professional development courses. The syllabus framework is recommended by the Swedish National Agency for Education and address what programming skills K-12 teachers in mathematics and technology should learn, and later use with their students.

In the third step, the game was designed and developed with the game development tool RPG Playground (RPGPlayground.com) where web-based role-playing games (RPGs) can be created and shared on the website.

Since the game is web-based it can be played on both computers and mobile devices. The game is built on the idea of educational escape rooms to facilitate motivation and learning through problem-solving (Hacke, 2019; López-Pernas et al., 2019). The aim of the game is to escape 10 rooms through problem-solving, where each room address a specific programming concept from the syllabus framework. In total, 5 programming concepts are addressed throughout the game. The first time in an easier challenge and the second time in a more difficult challenge (Table 1).

Table 1. Summary of content in *Escape with Python*-game

Room	Programming concept
01	Variables
02	Variables
03	Sequences
04	Sequences
05	Conditions
06	Conditions
07	Iterations
08	Iterations
09	Functions
10	Functions

The game has shorter dialogues with non-playable characters (NPCs) to learn programming concepts, the players’ knowledge is tested by trying to escape a room where that knowledge is required. In both the dialogues and challenges, metaphors and real Python-code are used to exemplify the programming concepts. The working title of the game is *Escape with Python* (Figure 1). Step 4 (demonstrate artefact) and step 5 (evaluate artefact) are addressed in the subsections: 3.1 *Data collection* and 3.2 *Data analysis*.



Figure 1. Screenshot from *Escape with Python* (By Niklas Humble in RPGPlayground.com)

3.1 Data collection

The game was demonstrated and evaluated by 32 students in K-12 education during the autumn semester of 2021 and spring semester of 2022. Data were collected through a questionnaire with a combination of Closed-ended and Open-ended questions to collect comments, feedback, and suggestions on the perception of playing the game, and how to improve it (Farrell, 2016; Hancock, Ockleford & Windridge, 2001). The collection process started with an open invitation for K-12 students to test and evaluate the game in a professional development course on programming for K-12 teachers in the autumn semester of 2021.

4 teachers answered the call and expressed interest in testing and evaluating the game with their students. The questionnaire was sent to the teachers, together with information about the study and an agreement form, to be distributed among students. 2 of the teachers were able to complete the test and evaluation of the game with their students and re-sent the questionnaires and agreement forms. The game was used as part of an

introduction to programming by one teacher, and together with students that had showed interest in programming by the other teacher.

In total 32 students tested and evaluated the game. 28 students in grade 9, 2 students in grade 8, 1 student in grade 7, and 1 student did not specify grade. The gender distribution of the students were 18 boys, 11 girls, and 3 did not specify gender. The most common context for testing and evaluating the game was during a lesson in technology (23), followed by mathematics (4) and programming (3). Students attend K-12 schools in the mid and north part of Sweden.

3.2 Data analysis

Answers to the Closed-ended questions were analysed in a spreadsheet document to calculate the average value of each question and the number of students that reported an increase or decrease in knowledge and interest development in programming by playing the game. In total there were 4 Closed-ended questions on perceived knowledge and interest in programming before and after playing the game. In each question, students were asked to grade their knowledge/interest in programming on a scale from 1 to 10 before and after playing the game.

Answers to the Open-ended questions were analysed in a thematic analysis to identify themes and group these into categories that answer the study's aim and research question (Clarke, Braun & Hayfield, 2015). A spreadsheet document was used, and themes were grouped into subcategories under each Open-ended question. The subcategories were then reorganised in the categories: Design factors for game graphics, Design factors for game mechanics, Design factors for game content, Design factors for learning and engagement. In total, there were 7 Open-ended questions on the perception of the game. Questions addressed the following topics: 1) look and graphics of the game, 2) interaction and mechanics of the game, 3) content and challenge of the game, 4) knowledge development of the game, 5) interest development of the game, 6) engagement of the game, and 7) other opinions and suggestions for improvements.

4. Results and analysis

Average value of Closed-ended questions showed no significant change in perceived knowledge or interest development in programming by playing the game. The average answered values of knowledge and interest in programming are about 4.6 both before and after playing the game (Table 2).

Table 2. Average value on a 1-10 scale for Closed-ended question

Question	Average value on 1-10 scale (rounded for 2 decimals)
How would you value your knowledge in programming before playing the game (where 1 is very low and 10 is very high)?	4.48 (31 answers)
How would you value your knowledge in programming after playing the game (where 1 is very low and 10 is very high)?	4.64 (31 answers)
How would you value your interest in programming before playing the game (where 1 is very low and 10 is very high)?	4.77 (30 answers)
How would you value your interest in programming after playing the game (where 1 is very low and 10 is very high)?	4.58 (31 answers)

However, analysing for increase and decrease of knowledge and interest in programming by playing the game, results are different (Table 3). 12 students valued that their knowledge in programming had increased by playing the game, while 5 students valued that their knowledge had decreased by playing the game, 14 students valued that their knowledge stayed the same by playing the game, and 1 student did not answer both questions. Regarding interest in programming, 7 students valued that their interests had increased by playing the game, while 6 students valued that their interest had decreased by playing the game, 17 students valued that their interest stayed the same by playing the game, and 2 students did not answer both questions.

Table 3. Perceived knowledge/interest development by playing the game

Knowledge / Interest	Increase	Decrease	Same
Perceived knowledge development by playing the game	12	5	14
Perceived interest development by playing the game	7	6	17

The Open-ended questions provide more detailed answers on the perceptions of the game and suggestions for improvements. Results of the thematic analysis are presented in sub-headings below.

4.1 Design factors for game graphics

Results of the open-ended questions show that students liked the retro-feel of the game and that it looked familiar to other games that they play. On the more critical side, some students suggested that there could be improvements to the overall design of the game. More details in the rooms, better graphics, background music, and voice acting were suggested. On the other side, some student appreciated the minimal design of the game and expressed that they liked that they could see what they were doing in the game. It was also suggested that player customisation should be added to the game, which would allow players to take control over how they were visually represented in the game (Quote 1).

“Good, the design was simple and you did not get overwhelmed. But you should be able to do customisation on the character that you play and design its’ look. Or, if there were other worlds that you could travel to.”

Quote 1. Student about graphical design and customisation of the game.

4.2 Design factors for game mechanics

The results further show that students perceived the gameplay as unclear or inconsequent throughout the game. For example, that an approach used for problem-solving in one room did not reoccur in the next (Quote 2). On the other hand, some students perceived that the game was easy to play and understand. Students also suggested that more freedom should be added to the gameplay. For example, players should be allowed to explore the game environment more freely and be able to write real code in the game, instead of selecting pre-written code. It was also expressed that dialogues with NPCs were appreciated but presented too slow.

“In some rooms you used those glowing things to do stuff but in other rooms you walked up to some character to do stuff. It feels a bit inconsistent.”

Quote 2. Student about gameplay and mechanics of the game.

4.3 Design factors for game content

Regarding content of the game, results show that some students perceived it to be too difficult while others perceived it to be too easy. This seems to relate to students’ prior experience of programming, which were varied. Results further show that the presentation of code and instructions in the game’s dialogues could be improved. There were too many dialogues to read, and code was difficult to understand when presented in a dialogue. Students also perceived that it was difficult to remember everything they needed to solve the problems in the rooms. Students further suggested that more excitement could be added to the game by applying a “life-system”, which would determine the numbers of chances that players would get to solve a problem (Quote 3). This would also address the problem pointed out by students that it was possible to “cheat” or guess your way through the game, since there were no repercussions for wrong answers or choices.

“I think that the game should have some sort of life-system. Because you can test all the answers without consequences. You don’t really have to think to finish the game.”

Quote 3. Student about the content and challenge of the game.

4.4 Design factors for learning and engagement

Students expressed that they found the game engaging and that they learned when game content related to school content or when the game was perceived as usable. Students expressed that they learned loops, being iterative, being accurate, and wanting to learn more about programming and creating their own games. Some students expressed that they perceive programming to be difficult and require too much work. This could be addressed with more in-game explanations and hints, which was also highlighted in the comments. Lastly, the results show a comparison between learning games and leisure games. Some students expressed that they were

not interested in programming or a game about programming, that it was not fun, or should be about something else than programming (Quote 4). Some students expressed that they liked the game, found it to be fun (for a learning game) but that they would not play it outside school.

“To some degree the game was fun. But I think it was very much the same over and over again. Maybe it should be about something else than programming. For example, about going on an adventure.”

Quote 4. Student about engagement in the game.

5. Discussion

A potential problem for the study's results were identified in the analysis of the closed-ended questions. Students that expressed a more critical or negative view of the game often reported a larger decrease in knowledge and interest in programming after playing the game. While students that expressed a more positive view on the game were more modest in their reported increase of knowledge and interest in programming after playing the game. This can be viewed in the presented results of the study where the average knowledge and interest in programming is almost the same before and after playing the game, while more students reported increase in knowledge and interest after playing the game than students reported decrease. Another potential problem with the closed-ended questions were that some students reported no increase in either knowledge or interest in programming with the motivation that they already liked programming and that the game therefore had no impact in that regard.

It should also be questioned whether one's knowledge can decrease by playing a game, since that would indicate that the game makes the players forget what they already know. Another interpretation of the reported decrease in knowledge is perhaps that those students did not like the game. There were also some indications in the analysis of the closed-ended questions that maybe all students did not understand how the grading scale worked. Some students were positive in the comments of the open-ended questions but had reported decrease in both knowledge and interest after playing the game.

The findings of the study contain both expected and unexpected results. That students favoured consequent gameplay or mechanics throughout the game was to be expected, since this allows the player to focus on the content and challenges of the game and not re-learning how it should be played. Similarly, it was expected that the students do not want the game content to be too difficult. An easy start, or low threshold, is preferred to build confidence and enable a sense of learning or skill progression. However, there are also some unexpected findings in the study's results. It was not expected that students would like the minimal design of the game, although this was not the case for all students, and refer to it as “retro design” given their young age. However, this could be explained by the access to mobile games that often share this design. Further, it was unexpected that students assessed the game in relation to learning games and leisure games. This could be viewed as an indication that the students are accustomed to learning games in school.

An important part of games is challenge, and a progression of challenge, to keep the player in the flow channel. With flow and the flow channel defined by Csikszentmihalyi (1975) as state of peaked enjoyment and energetic focus, in-between border states of anxiety and boredom. The recommendation from UX designers is that game designers should consider the flow state as the optimum user experience and try to keep players in the flow channel (Neal, 2012). However, in this case with programming techniques as an increasingly challenging content, the gameplay challenges must not simultaneously increase. For players without earlier programming and gaming skills, this could lead away from the flow channel towards the state of anxiety.

6. Conclusion

The study shows varying results on students' perceptions of the game, some students like the design and content of the game and some students do not. With the same division, some students perceive that the game engage them and facilitate learning, while other students do not. The conclusion of the study is that there are several important design factors regarding graphics, mechanics, content, learning and engagement to consider when developing a digital escape room game on computer programming. Style and graphics of the game should be familiar to other games that students play. However, this does not necessarily imply state of the art graphics. Many games today, especially mobile, have a minimal and retro design. Gameplay and mechanics should be consistent throughout the game, or at least in the beginning of the game, especially if the game content is directed towards novices. Content, or challenges, of the game should be presented clearly and exciting, with appropriate level of difficulty for the target group. To facilitate learning and engagement, the game should be

perceived as usable by the students. For example, by relating to school content or developing skills of interest and relevance for the students. Lastly, the authors suggest incorporating high level of freedom in the game to facilitate engagement. For example, freedom for player customisation and freedom in world exploration. The study suggests that these design factors, if properly addressed, can support in lowering the threshold for educational escape rooms on programming and raising the ceiling for learning and engagement.

7. Limitations and future research

This study was conducted with a relatively simple game development tool and a limited number of game testers. In the next iteration of game development, this prototype will be rebuilt with a new and more professional game development tool and tested on a larger scale. To test the game's effectiveness for learning programming it would further be desirable to test the game related to learning objectives, for example with the use of a pre- and post-test study design. Another interesting idea would be to include students in the game development process through participatory design. This could potentially allow for early detection of mistakes in the game design, such as it being too text-heavy.

References

- Admiraal, W., Post, L., Guo, P., Saab, N., Makinen, S., Rainio, O., ... & Danford, G. (2019). Students as future workers: Cross-border multidisciplinary learning labs in higher education. *International Journal of Technology in Education and Science*, 3(2), 85-94.
- Allison, J. (2021, November). The Importance of Context: Assessing the Challenges of K-12 Computing Education through the Lens of Biggs 3P Model. In *21st Koli Calling International Conference on Computing Education Research* (pp. 1-10).
- Borrego, C., Fernández, C., Blanes, I., & Robles, S. (2017). Room escape at class: Escape games activities to facilitate the motivation and learning in computer science. *JOTSE*, 7(2), 162-171.
- Chetty, J., & Barlow-Jones, G. (2012). Integrating teaching-and-learning techniques for novice computer programming students. In *Global Learn* (pp. 142-146). Association for the Advancement of Computing in Education (AACE).
- Clarke, V., Braun, V., & Hayfield, N. (2015). Thematic analysis. *Qualitative psychology: A practical guide to research methods*, 222, 248.
- Csikszentmihalyi, M. (1975). *Beyond boredom and anxiety: The experience of play of play in work and games*. San Francisco, CA: Jossey-Bass.
- Engström, E., Storey, M. A., Runeson, P., Höst, M., & Baldassarre, M. T. (2020). How software engineering research aligns with design science: a review. *Empirical Software Engineering*, 25(4), 2630-2660.
- Farrell, S. (2016, September 25). 28 Tips for creating great qualitative surveys. Retrieved 2022, April 1 from <https://www.nngroup.com/articles/qualitative-surveys/>.
- Gomes, A., & Mendes, A. J. (2007). An environment to improve programming education. In *Proceedings of the 2007 international conference on Computer systems and technologies* (pp. 1-6).
- Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International journal of child-computer interaction*, 16, 77-84.
- Hacke, A. (2019, November). Computer science problem solving in the escape game "Room-X". In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 281-292). Springer, Cham.
- Hancock, B., Ockleford, E., & Windridge, K. (2001). *An introduction to qualitative research*. London: Trent focus group.
- Hawi, N. (2014). Learning programming: a model emerging from data. *International Journal of Computer Applications*, 100(4).
- Heintz, F., Mannila, L., Nordén, L. Å., Parnes, P., & Regnell, B. (2017, November). Introducing programming and digital competence in Swedish K-9 education. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 117-128). Springer, Cham.
- Hellerstedt, A., & Mozelius, P. (2019). Game-based learning: A long history. In *Irish Conference on Game-based Learning 2019, Cork, Ireland, June 26-28, 2019* (Vol. 1).
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.
- Humble, N., Mozelius, P., & Sällvin, L. (2020). Remaking and reinforcing mathematics and technology with programming—teacher perceptions of challenges, opportunities and tools in K-12 settings. *The international journal of information and learning technology*.
- Humble, N., Mozelius, P., & Sällvin, L. (2021). You can't Escape Learning, but Maybe you can get out of the room!: Game-based Learning for Programming Education. In *15th European Conference on Game Based Learning (ECGBL 2021), Brighton, Uniyted Kingdom, September 23-24, 2021*. (pp. 359-367). Academic Conferences International Limited.
- Hussein, M. H., Ow, S. H., Cheong, L. S., Thong, M. K., & Ebrahim, N. A. (2019). Effects of digital game-based learning on elementary science learning: A systematic review. *IEEE Access*, 7, 62465-62478.
- Johannesson, P., & Perjons, E. (2014). *An introduction to design science* (Vol. 10, pp. 978-3). Cham: Springer.
- Lédeczi, Á., Grover, S., Catete, V., & Broll, B. (2021). Beyond CS Principles: Bringing the Frontiers of Computing to K12. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 1379-1379).

- López-Pernas, S., Gordillo, A., Barra, E., & Quemada, J. (2019). Examining the use of an educational escape room for teaching programming in a higher education setting. *IEEE Access*, 7, 31723-31737.
- López-Pernas, S., Gordillo, A., Barra, E., & Quemada, J. (2021). Comparing Face-to-Face and Remote Educational Escape Rooms for Learning Programming. *IEEE Access*, 9, 59270-59285.
- Malliarakis, C., Shabalina, O., & Mozelius, P. (2021). Can you Escape from Dr. Tom Cat's Lab? Educational Escape Rooms with Scientists, Riddles and Serious Games as Learning Tools. In *European Conference on Game Based Learning, ECGBL 2021, Brighton, United Kingdom, September 23-24, 2021*. (Vol. 15). Academic Conferences and Publishing International Limited.
- Marcolino, A. S., & Barbosa, E. (2017, October). A survey on problems related to the teaching of programming in Brazilian educational institutions. In *2017 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.
- Miljanovic, M. A., & Bradbury, J. S. (2018). A review of serious games for programming. In *Joint international conference on serious games* (pp. 204-216). Springer, Cham.
- Neal, M. (2012). Creating and maintaining psychological flow state in augmented reality applications. In *Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17.
- Pereira, F. D., Oliveira, E., Cristea, A., Fernandes, D., Silva, L., Aguiar, G., ... & Alshehri, M. (2019, June). Early dropout prediction for programming courses supported by online judges. In *International conference on artificial intelligence in education* (pp. 67-72). Springer, Cham.
- (The) Royal Society (2017, November). After the reboot: computing education in UK schools. Retrieved 2022, Mars 31 from <https://royalsociety.org/~media/events/2018/11/computing-education-1-year-on/after-the-reboot-report.pdf>.
- Seralidou, E., & Douligeris, C. (2021). Learning programming by creating games through the use of structured activities in secondary education in Greece. *Education and Information Technologies*, 26(1), 859-898.
- Silva, J., & Silveira, I. (2020). A Systematic Review on Open Educational Games for Programming Learning and Teaching. *International Journal of Emerging Technologies in Learning (IJET)*, 15(9), 156-172.
- Szabo, C., Sheard, J., Luxton-Reilly, A., Becker, B. A., & Ott, L. (2019, November). Fifteen years of introductory programming in schools: a global overview of K-12 initiatives. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1-9).
- Thompson, K. (1986). Retrograde analysis of certain endgames. *J. Int. Comput. Games Assoc.*, 9(3), 131-139.
- Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education*, 162, 104083.
- Turing, A. M. (1953). Digital computers applied to games. *Faster than thought*.
- Urbietá, A. S., & Peñalver, E. A. (2019). Escaping from the English classroom. Who will get out first?. *Aloma: revista de psicologia, ciències de l'educació i de l'esport Blanquerna*, 37(2), 83-92
- Vankúš, P. E. T. E. R. (2005). History and present of didactical games as a method of mathematics' teaching. *Acta Didactica Universitatis Comenianae-Mathematics*, 5, 53-68.
- Veldkamp, A., van de Grint, L., Knippels, M. C. P., & van Joolingen, W. R. (2020). Escape education: A systematic review on escape rooms in education. *Educational Research Review*, 31, 100364.
- Yayon, M., Rap, S., Adler, V., Haimovich, I., Levy, H., & Blonder, R. (2019). Do-It-Yourself: Creating and Implementing a Periodic Table of the Elements Chemical Escape Room. *Journal of Chemical Education*, 97(1), 132-136.