International Journal of Research in E-learning
Vol. 11 (2), 2025, pp. 1–22

DOI 10.31261/IJREL.2025.11.2.03

Maria Wisniewska
Public General High School of the Lodz University of Technology, Poland
https://orcid.org/0009-0001-3238-8674

Zbigniew Wisniewski
Lodz University of Technology, Poland
https://orcid.org/0000-0003-0066-9321

A Scratch-Based Simulation of Virus Spread as a Constructionist E-Learning Project

Abstract

This article presents a qualitative case study of a Scratch-based simulation project created by a primary school student during a period of remote learning. The study explores how designing a simple agent-based model in a visual programming environment can support the development of digital and computational competences. The analysis draws on three sources of data: the Scratch artefact, competition documentation, and a retrospective semi-structured interview with the project's author. The findings indicate that constructing the simulation helped the student develop key computational practices – such as decomposition, iterative refinement, and problem-solving – and enabled her to explore causal relationships within a simplified model. The project demonstrates how accessible programming environments can facilitate constructionist learning by engaging learners in designing and testing executable artefacts. The results further suggest that simulation-based projects can enrich remote and hybrid education by providing opportunities for active experimentation and reflective analysis. The study's main limitation is its single-case design, which narrows the scope for generalisation. Further comparative and longitudinal research is needed to assess wider educational impact of student-created simulations.

K e y w o r d s: constructionism, Scratch, remote learning, computational thinking, digital competences, civic competences

Introduction

The period of remote education in 2020–2021 rapidly accelerated the integration of digital technologies into everyday school practice, and created fertile ground for projects built around programming and computer-based simulations. One example of such initiatives was a nationwide IT competition organized by Lodz University of Technology, in which students prepared applications or presentations in Scratch that addressed contemporary social issues. The competition task combined elements of computational thinking, problem-solving, and social reflection, offering a particularly clear illustration of constructionist, project-based learning.

Within this context, the project examined in this case study emerged – a simulation illustrating the spread of a virus in a self-service store, created by the article's co-author, then a fourteen-year-old student participating in remote schooling. She chose to develop the project in the form of a simulation-game, blending agent-based modeling, interactivity, and randomness. This design allowed users to test various scenarios and observe how changes in model parameters affected infection dynamics.

The purpose of this article is to analyze the project from pedagogical and technological perspectives, with particular attention to how designing a simulation in a visual programming environment can foster computational thinking, digital competences, and an inquiry-oriented approach to learning. The case study also contributes to the broader discussion on the role of simulation-based projects in e-learning and on how simple programming environments can support the exploration of complex social phenomena.

The presented case offers not only a view of the final artefact but also a detailed look at how a student constructs a simulation, what decisions she makes along the way, and which features of the visual programming environment support her computational and project-based practices. This case study introduces three elements that distinguish it from Scratch-based pandemic projects typically described in the literature: (1) it documents the entire process of simulation design – from the initial concept, through iterative refinements, to sharing the artefact with other users; (2) it represents an independently developed, simplified agent-based model of a social phenomenon, whereas most educational Scratch projects focus on animations, simple games, or introductory programming tasks; and (3) it draws on three complementary data sources – the programming artefact, competition documentation, and a retrospective interview – allowing the code structure to be linked directly with the student's narrative and the institutional context. Such triangulation is rarely discussed in studies of student-created Scratch projects.

1. Theoretical Background

The COVID-19 pandemic accelerated the adoption of digital technologies in education, drawing renewed attention to learning models that foreground student activity, independent experimentation, and the creation of digital artefacts. In this context, three interrelated perspectives have gained particular relevance: constructionism, computational thinking, and learning through simulations and project-based activities.

Constructionism and Learning Through Making

Constructionism (Papert, 1991, 1993) posits that learners develop knowledge most effectively when they create tangible artefacts – programs, models, animations – that externalize and refine their understanding of a phenomenon. The act of making becomes a cognitive process in itself, involving hypothesis building, testing, and reflection. Within the setting of remote education, constructionism provides a framework that supports autonomy, creativity, and self-regulation, enabling learners to explore and create artefacts in digital environments such as Scratch (Levin et al., 2025; Hodges et al., 2020; Bozkurt & Sharma, 2020).

Computational Thinking as a Design-Supporting Competence

Computational thinking (Wing, 2006) encompasses the ability to decompose problems, identify patterns, design algorithms, and engage in abstract reasoning. In the framework proposed by Brennan and Resnick (2012), it consists of three components: programming practices (e.g., iteration, testing), programming concepts (such as variables, loops, and conditionals), and cognitive dispositions including persistence and curiosity. Designing simulations in Scratch naturally develops these elements, as it requires learners to blend system logic with experimentation, iterative refinement, and independent problem-solving.

Simulations and Project-Based Learning as Tools for Inquiry

Using simulations as educational tools allows learners to explore phenomena by manipulating parameters and observing resulting patterns. Following the inquiry learning model (de Jong & van Joolingen, 1998), students formulate hypotheses, test them in a simulated environment, and interpret the outcomes. Even simplified agent-based simulations allow learners to examine causal relationships and observe emergent behaviors. When combined with constructionist principles, designing a simulation becomes a form of learning through inquiry and project-based exploration.

Visual Technologies and Learning in Digital Environments

Visual programming environments such as Scratch lower the entry threshold for programming and modeling, enabling students to translate intuitive ideas into executable code. In educational practice, they are recognized as tools that foster creativity, iterative work, and the sharing of results (Resnick, 2017). In the context of remote learning, these environments gain additional importance – they allow students to undertake projects that integrate social, technical, and civic themes while maintaining the practical dimension of experimentation.

In the case discussed here, Scratch served as a natural environment for developing such skills: the student designed, tested, and shared her work with others. The DigCompEdu framework (Punie, 2017) underscores the value of creative and critical uses of technology in teaching, emphasizing autonomy and collaboration – the elements clearly visible in this project.

Integrating the Three Perspectives

This article adopts an interpretative framework that brings together the three areas described above:

- constructionism, understood as the philosophy and method of "learning through making";
- computational thinking, viewed as the set of competences employed in designing the model;
- simulation-based learning, understood as a means of exploring social and epidemiological phenomena.

This integrated perspective makes it possible to analyze the project both as a technical artefact and as an educational tool that fosters reflection, digital competences, and inquiry-based practices.

2. Case Study Context

In the 2020/2021 school year, when the education system operated almost entirely in a remote mode, students worked with limited contact with teachers and peers, which required greater independence in planning and completing learning tasks. Research shows that the pandemic acted as a catalyst for the rapid digital transformation of schools and stimulated interest in pedagogical models emphasizing learner activity, creativity, and the construction of digital artefacts (Hodges et al., 2020; Bozkurt & Sharma, 2020). In many countries, project-based work and technology-supported initiatives were among the few elements that positively influenced students' motivation and engagement during remote learning (European Schoolnet, 2021; Konin.pl, 2020).

Against this backdrop, the nationwide InfoSukces IT competition organized by Lodz University of Technology created an opportunity for students to develop independent digital projects. The competition task – designing a Scratch presentation or application related to pandemic-specific phenomena – aligned with constructionist principles that promote learning through the creation of artefacts (Papert, 1991; Ackermann, 2001). The evaluation criteria, which included originality, logical coherence, and educational value, reflected broader trends in combining creativity, reflective activity, and computational thinking in school-based projects (Brennan & Resnick, 2012; Levin et al., 2025).

The aim of the student's project was to enable users to explore the relationships between human behavior and the dynamics of virus transmission in a self-service store. This was achieved by designing an interactive simulation that allowed users to modify social conditions within the store environment and observe how these changes influenced the spread of the virus. The author of the analyzed project was a fourteen-year-old student attending school remotely, which meant that the work had to be planned and carried out independently – a pattern consistent with research emphasizing the increased importance of self-regulation during remote learning (Dumont et al., 2021; Długosz, 2022). Working in Scratch enabled her to combine programming knowledge with intuitive insights about social behaviors observed during the pandemic. The process of translating personal experiences into an educational model aligns with constructionist approaches that view artefacts as externalizations of learners' understanding of the world (Papert, 1993; Resnick, 2017).

The project received second place in the competition, confirming both its technical quality and conceptual strength. The student published the simulation on the Scratch platform, allowing other users to experiment with variables and explore the outcomes of different scenarios – a practice aligned with the constructionist idea of learning by sharing artefacts (Resnick, 2017) and with inquiry learning, which emphasizes experimentation and hypothesis testing in simulated environments (de Jong & van Joolingen, 1998; Fukuda et al., 2022).

3. Research Methodology

This study was conducted within a qualitative paradigm using a case study design, which makes it possible to examine a complex educational phenomenon within its natural context (Crowe et al., 2011). A single-case approach was selected, with embedded units of analysis comprising the Scratch application, the competition documentation, and the retrospective interview data. Given that, the boundaries between the phenomenon under study (the simulator project) and its context (the pandemic and remote learning) were inherently blurred, this method was particularly appropriate (Yin, 2014).

Data Sources and Their Role in the Analysis

Three categories of data were used in the analysis, as summarized in Table 1. Their triangulated use allowed for the comparison of information from multiple sources and strengthened the credibility of the findings (Yin, 2014).

Table 1.
Data sources and their role in the analysis

Type of data	Description	Role in the analysis
Artefact – Scratch project	Program code, system logic, sprite structure, simulation mechanics	Reconstruction of the model design process; identification of design decisions and elements of computational thinking
Competition documents	InfoSukces regulations, evaluation criteria, submission requirements	Situating the project within its institutional context; analysing constraints and expectations shaping design choices
Retrospective semi-structured interview	A 30-minute recorded and transcribed interview with the project author	Eliciting information about motivations, strategies, project iterations, and reflections on the learning process

Source: Own work.

Artefact Analysis

The artefact analysis focused on the structure of the Scratch code, the relationships between modules, and the implementation of agent movement, interactions, and randomness. Particular attention was given to design decisions reflecting computational thinking practices such as decomposition, iteration, and the use of loops and conditionals (Brennan & Resnick, 2012). Key technical components of the program are presented in Appendix B.

Document Analysis

The competition documentation was examined with regard to its objectives, project requirements, and evaluation criteria. This analysis made it possible to understand the institutional expectations that may have influenced both the scope and form of the developed simulation.

Retrospective Interview

The interview with the project author followed a semi-structured format, enabling flexible exploration of themes related to motivation, conceptual development, and the progression of programming iterations (Kvale, 2007). The analysis was conducted using thematic coding, which helped identify key areas: motivations and project context, the workflow and creative process, strategies for coping with difficulties, an examination of design choices, and broader reflections on the significance of the project. The full list of interview questions is provided in Appendix A.

Process Tracing

Elements of process tracing were also employed to reconstruct the sequence of design decisions – from the initial concept, through iterative coding and testing, to the final version of the simulation (Collier, 2011). This approach made it possible to capture the mechanisms underlying each stage of artefact creation and to understand how they contributed to the student's developing skills.

4. Results and Discussion

4.1. Analysis of the Artefact and Design Structures

An analysis of the Scratch application shows that the student designed a simulation in which agents (store customers) move autonomously, interact with one another, and generate events that lead to the spread of the virus. The logical layer of the model reveals the use of key computational thinking practices: decomposing the process into modules (movement, infections, interactions), employing loops and conditionals, operating with variables, and introducing randomness (Brennan & Resnick, 2012; Wing, 2006).

The project "Coronavirus in the Store" (Fig. 1) was developed in Scratch 3.0. The structure of the program is built around four main modules corresponding to the logic of the simulation: customer (agent) movement, infections and virus particles, interactions and infection conditions, and the user panel with result visualization. The key technical components of the program are presented in Appendix B.

The simplified agent-based simulation created by the student reflects characteristic features of emergent models, in which global outcomes (e.g., the number of infections) arise from the collective behaviour of individual agents operating according to simple rules (Railsback & Grimm, 2019; Ormazábal et al., 2021). The choice of such a structure suggests an intuitive recognition of the properties of complex social systems on the part of the student.

The program code embodies computational thinking through:

- the decomposition of the epidemiological phenomenon into agents, movement, and interactions;
- the use of loops, variables, conditionals, and randomness as tools for representing processes;
- iterative testing and refinement of the code consistent with engineering practices.

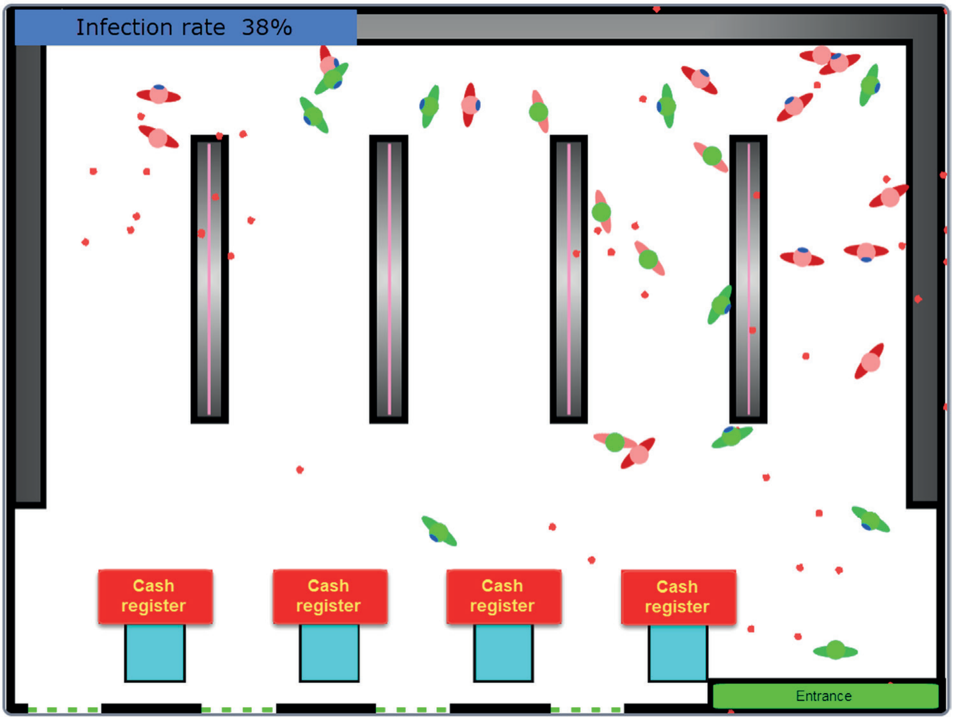

Figure 1. View of the simulation environment.

Source: Own work.

From a constructionist perspective, creating this program constituted a form of learning through action: the author explored epidemiological, computational, and social concepts by transforming them into a material artefact. Each block of code became a locus of reflection on reality, while choosing a parameter value was, in effect, a research decision.

From the standpoint of educational simulations, the program implements core principles of inquiry-based learning (de Jong & van Joolingen, 1998): the user can manipulate parameters, observe the outcomes, and draw inferences. An additional strength of the project is that it was conceived from the outset as a game, which elicits specific user behaviours and stimulates cognitive and exploratory engagement. The user independently observes causal relationships, confronting their own decisions (e.g., number of people in the store, assigning masks to healthy or infected customers) with the resulting epidemiological effects. This distinctive approach to the competition task aligns with inquiry-based simulation learning, in which hypothesis testing leads to conclusions grounded in model results (Zhang, 2025). The model itself functions as a constructionist artefact: in designing it, the student encoded her social and health-related intuitions into the logic of the simulation.

The artefact analysis thus confirms that the project is a coherent example of multi-agent modelling in an educationally friendly environment and is fully aligned with the ideas of digital constructionism.

4.2. Document Analysis

The analysis of the InfoSukces competition regulations, issued by Lodz University of Technology and the Regional Board of Education in Łódź, situates the project within a broader institutional and didactic context. The competition was national in scope and targeted students in grades 7–8 of primary school. Its aims included developing students' interest in computer science, promoting logical thinking, and encouraging creative use of information technologies. The document analysis highlighted several key aspects:

1. Aims and rationale of the competition: InfoSukces is consistent with educational constructionism in that it promotes the creation of original digital artefacts rather than the reproduction of content. The evaluation criteria (originality, aesthetics, functional logic, and use of Scratch) placed strong emphasis on creativity, independence, and the practical application of computing skills.
2. Theme and requirements: in the 2020/2021 edition, the pandemic theme granted the projects a social and emotional dimension. The task was to create a project that addressed the problem of the pandemic in the form of an educational program or presentation.
3. Evaluation system: projects were assessed according to three main criteria: programming effort, quality of visual design, and originality of the idea. This balanced focus on technical, visual, and conceptual aspects encouraged design thinking rather than merely checking for technical correctness.

4.3. Analysis of the Semi-Structured Interview Transcript

The interview had a retrospective character, was conducted in person, and lasted 30 minutes. Its analysis made it possible to reconstruct the student's learning process in alignment with the stages embedded in the research instrument. The author indicated that her main motivation for creating the simulator was the need for an intellectual challenge and the desire to apply skills acquired earlier in a programming course.

During the project, she employed practices consistent with the idea of learning by making. She described planning her actions independently, using loops and randomness in the code, and iteratively testing the program while adding new functionalities such as improved graphics and sound effects. The process was highly autonomous – she analysed errors on her own and searched for solutions. This supports the claim that a constructionist approach can foster both computational thinking and persistence in problem-solving (Brennan & Resnick, 2012).

The student also reported that, although the pandemic period was emotionally difficult, the work on the project itself was not associated with negative emotions; on the contrary, it was a source of satisfaction and a sense of agency. She noted that programming demands concentration and solitary work, which meant that isolation did not hinder but rather naturally accompanied the creative process. This observation echoes Gee's (2003) contention that virtual environments can provide "safe spaces" for experimentation and knowledge construction under conditions of social distancing.

On the social level, the author acknowledged that the simulator had an educational purpose and reflected knowledge circulating in the media at the time – concerning masks, physical distancing, and group size. The goal was to illustrate that these factors have a tangible impact on the pace of infections. She also emphasised that she did not aim to achieve scientific fidelity to real epidemiological processes, but rather to build a "conceptual schematic". Despite this awareness, the project had a social dimension: it was shared with peers, enabling joint experimentation and conversations about the consequences of pandemic-related behaviours.

The choice of a self-service store as the simulation environment was motivated by its universality and the relative ease of modelling processes taking place there. The selection of parameters (number of people, distance, masks) reflected their prominence in public discourse, pointing to an intuitive transfer of social knowledge into programming logic.

4.4. Process Tracing

The retrospective interview showed that the project evolved in a cyclical manner, consistent with the iterative model described in constructionist literature: create–test–improve (Papert, 1991; Resnick, 2017). The student began with a sketch-like model and then systematically extended it by adding new mechanisms, such as masks, different types of customers, random events, and an infection counter.

Each iteration was driven by observing how the model behaved and identifying problems. For example:

- when infections spread too quickly, she introduced a mask protection coefficient;
- when agent movement appeared chaotic, she added direction planning and collision constraints;
- when interpreting the results proved difficult, she expanded the counters and simplified the user interface.

This iterative approach corresponds to constructionist practices in which learners test hypotheses by modifying an artefact and learning from errors (Harel &

Papert, 1991; Levin et al., 2025). At a certain point, the project reached a level of complexity that exceeded the technical capabilities of the environment, which the student interpreted as evidence of her own progress. This reflection suggests that technological limits can act as a learning milestone – a moment when learners become aware of both their competencies and the constraints of the tool (Wing, 2006).

Based on the analysis of the code and the student's narrative, a set of actions corresponding to the three dimensions of computational thinking (CT) proposed by Brennan and Resnick (2012) was identified:

Computational concepts

The student built a model based on variables, random functions, conditionals, and loops controlling agent behaviour. This not only allowed her to construct the logic of the simulation but also to understand parametric dependencies – such as the impact of the number of customers or mask usage on system behaviour.

Computational practices

The project involved typical practices such as iterative modification, debugging, scenario testing, and exploring edge cases. The test–refine cycle, central to constructionism, was particularly evident.

Computational perspectives

The student's statements indicate that the project enabled her to consciously connect social observations with model logic – what the literature describes as the development of model-based reasoning in simulation environments (de Jong & van Joolingen, 1998; Wen et al., 2020). She noted that she understood "why the model behaves in one way and not another" and could observe the consequences of parameter changes, which supported her understanding of causal relationships.

An examination of how empirical data were linked to the findings shows that:

The artefact (code and program logic)

- made it possible to determine which elements of the model required specific programming and logical practices;
- enabled the identification of structures that support CT, such as loops, variables, conditionals, and events.

The interview

- confirmed that code iterations were conscious educational decisions (e.g., "I changed the model because infections were rising too fast");
- revealed the student's motivations and her understanding of her own learning process.

The competition documents

- clarified why the student chose a simulation as the project format (criteria of originality and educational value);
- enriched the analysis with the institutional context.

Triangulation thus made it possible to compare what the student declared with what she actually encoded, thereby strengthening the credibility of the results (Yin, 2014).

4.5. Comparison with Previous Research

The findings of this case study are consistent with the literature indicating that project-based and constructionist approaches can be effective even under remote learning conditions. The student worked independently, demonstrating high levels of motivation and engagement, which aligns with studies showing that, during the pandemic, students perceived projects as one of the most engaging forms of work (Konin.pl, 2020; European Schoolnet, 2021). Similar conclusions have been drawn by teachers involved in eTwinning projects, who reported that project work improved the organization of teaching and increased the attractiveness of learning materials (Kaya & Bagçeci, 2025).

In line with research on remote programming education, the case shows that projects implemented in visual environments support the development of digital competences and computational thinking. The student developed both programming concepts and iterative design practices, in accordance with the theoretical framework proposed by Brennan and Resnick (2012). These results are consistent with experimental studies demonstrating that project-based approaches to teaching programming can be effective online, provided that students have access to appropriate tools and tasks (Amnouychokanant et al., 2021). European research reviews also indicate that distance teaching of computer science requires close collaboration among teachers and deliberate use of digital media (Skaraki & Kolokotronis, 2022).

The findings also resonate with broader observations on remote education, which emphasize that active, learner-centred methods can counteract declines in motivation and learning quality frequently reported during lockdowns (Konin.pl, 2020; Długosz, 2022). Projects grounded in real-world problems enhanced students' sense of agency, which has been identified as a key factor in mitigating typical challenges associated with distance learning. European initiatives such as eTwinning had already shown that digital projects can foster not only technical skills but also collaboration, creativity, and problem-solving (Saab et al., 2011; Kaya & Bagçeci, 2025). Our case confirms that visual environments such as Scratch can serve as effective tools for developing these competences even under conditions of full isolation.

At the same time, several differences between this case and previous studies should be noted. First, the project emerged from strong internal motivation and the structural support of a competition, whereas many pandemic-era studies highlight students' difficulties with autonomy and sustained engagement in online work, especially among younger learners (Czerniewicz et al., 2020; Dumont et al., 2021). Second, most existing research on educational projects is descriptive in nature, and there is a lack of experimental evidence demonstrating their superiority over traditional methods in school settings (Skaraki & Kolokotronis, 2022). Third, relatively few studies examine the role of post-project reflection, even though research from other domains – including medical education – suggests that debriefing may be crucial for the durability of learning outcomes.

In summary, the results of this study align with dominant trends pointing to the substantial value of project-based learning and constructionism in remote education. At the same time, they underscore the need for further research involving larger groups of students and diverse contexts, as well as longitudinal analyses, to determine the extent to which the observed effects are generalizable or are specific to the conditions of a single case (Długosz, 2022).

5. Synthesis of Findings and Key Conclusions

The analysis of the interview, competition documentation, and the programming artefact indicates that designing the simulation in Scratch functioned as an example of learning through making, fully aligned with constructionist principles. The student demonstrated the ability to independently plan subsequent stages of work, solve technical problems, and iteratively refine the model – skills reflected both in the structure of the code and in her descriptions of the project's development process. These practices correspond to core elements of computational thinking, such as decomposition, testing, and step-by-step improvement.

The findings also show that even simple programming environments can meaningfully support the development of digital and computational competences when the project presents learners with authentic challenges and enables them to explore phenomena through simulation (see also: Montiel & Gómez-Zermeño, 2021). Working with an agent-based model allowed the student to engage with causal and parametric relationships, encouraging analysis and reflection on how her design functioned – without attempting to replicate real epidemiological processes.

Practical Recommendations

In light of the evidence gathered, the study suggests that simulation-based projects can be a valuable component of remote and hybrid learning. The creation of a functioning artefact fostered technical, logical, and organisational skills, while also cultivating habits of iterative work. Such forms of activity may increase student engagement and participation in the learning process. Moreover, incorporating structured reflection and discussion after simulation-based experiments (debriefing) is advisable, as research in medical education shows that this phase strengthens learning outcomes (Elendu et al., 2024). These conclusions align with previous analyses emphasising the utility of project-based approaches in digital learning environments.

Limitations

The single-case nature of the study limits the generalisability of its findings. The project represents an individual example rather than a pattern that can be assumed to apply to all student-created simulations. The absence of quantitative data – such as user statistics or system logs – prevents a fuller assessment of how the project may have influenced learners' attitudes or behaviours. Additionally, the retrospective interview may be subject to memory bias, and the simulation itself, despite its didactic value, did not account for many real epidemiological factors, which constrains its scientific precision – though achieving scientific accuracy was not its intended purpose.

Directions for Future Research

The findings point to the need for expanding research to include a greater number of student-created simulations across diverse school and regional contexts, allowing for comparative analyses of design strategies and educational outcomes. Future work could incorporate quantitative studies, including experimental designs in which classes engaging in simulation tasks are compared with control groups – in order to assess the impact of such activities on digital competences and conceptual understanding. Longitudinal studies would also be valuable to determine whether experience with simulation design influences later achievement in STEM subjects. Another promising direction involves integrating emerging technologies – such as artificial intelligence, machine learning, or real-world datasets – to explore their potential impact on learner engagement and learning effectiveness.

References

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference? http://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf (Retrieved: 2026-01-23)

Amnouychokanant, V., Boonlue, S., Chuathong, S., & Thamwipat, K. (2021). Online learning using block-based programming to foster computational thinking abilities during the COVID-19 pandemic. *International Journal of Emerging Technologies in Learning, 16*(13), 227–247. https://doi.org/10.3991/ijet.v16i13.22591 (Retrieved: 2026-01-23)

Bozkurt, A., & Sharma, R. C. (2020). Emergency remote teaching in a time of global crisis due to Coronavirus pandemic. *Asian Journal of Distance Education, 15*(1), i–vi. https://zenodo.org/records/3778083 (Retrieved: 2026-01-23)

Bozkurt, A., & Sharma, R. C. (2021). In pursuit of the right mix: Blended learning for augmenting, enhancing, and enriching flexibility. *Asian Journal of Distance Education, 16*(2), i–vi. https://doi.org/10.5281/zenodo.5827159 (Retrieved: 2026-01-23)

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada, April 13–17, 2012. http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf (Retrieved: 2026-01-23)

Collier, D. (2011). Understanding process tracing. *PS: Political Science & Politics, 44*(4), 823–830. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1856702 (Retrieved: 2026-01-23)

Crowe, S., Cresswell, K., Robertson, A., Huby, G., Avery, A., & Sheikh, A. (2011). The case study approach. *BMC Medical Research Methodology, 11*, 100. https://link.springer.com/article/10.1186/1471-2288-11-100 (Retrieved: 2026-01-23)

Czerniewicz, L., Agherdien, N., Badenhorst, J., Belluigi, D., Chambers, T., Chili, M., ... & Wissing, G. (2020). A wake-up call: Equity, inequality and Covid-19 emergency remote teaching and learning. *Postdigital Science and Education, 2*(3), 946–967. https://doi.org/10.1007/s42438-020-00187-4 (Retrieved: 2026-01-23)

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*(2), 179–201. https://doi.org/10.3102/00346543068002179 (Retrieved: 2026-01-23)

Dlugosz, P. (2022). Zdalne nauczanie wśród uczniów szkoły podstawowej na obszarach peryferyjnych w Europie Środkowo-Wschodniej [*Remote learning among primary school students in peripheral areas of Central and Eastern Europe*]. *Kwartalnik Naukowy Edukacja [Scientific Quarterly Education], 3*(162), 75–93.

Dumont, G., Ni, A. Y., Van Wart, M., Beck, C., & Pei, H. (2021). The effect of the COVID pandemic on faculty adoption of online teaching: Reduced resistance but strong persistent concerns. *Cogent Education, 8*(1), Article 1976928. https://doi.org/10.1080/2331186X.2021.1976928 (Retrieved: 2026-01-23)

Elendu, C., Amaechi, D. C., Okatta, A. U., Amaechi, E. C., Elendu, T. C., Ezeh, C. P., & Elendu, I. D. (2024). The impact of simulation-based training in medical education: A review. *Medicine, 103*(27), e38813. https://doi.org/10.1097/MD.0000000000038813 (Retrieved: 2026-01-23)

European Schoolnet. (2021). *eTwinning Monitoring Report 2021.* European Commission. https://op.europa.eu/en/publication-detail/-/publication/b8af9e2a-6144-11ec-9c6c-01aa75ed71a1/language-en (Retrieved: 2026.01.23)

Fukuda, M., Hajian, S., Jain, M., Liu, A. L., Obaid, T., Nesbit, J. C., & Winne, P. H. (2022). Scientific inquiry learning with a simulation: Providing within-task guidance tailored to learners' understanding and inquiry skill. *International Journal of Science Education, 44*(6), 1021–1043. https://doi.org/10.1080/09500693.2022.2062799

Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment, 1*(1), 20. https://doi.org/10.1145/950566.950595 (Retrieved: 2026-01-23)

Harel, I., & Papert, S. (1991). *Constructionism*. Ablex Publishing.

Hodges, C., Moore, S., Lockee, B., Trust, T., & Bond, A. (2020). The difference between emergency remote teaching and online learning. *EDUCAUSE Review*. https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning (Retrieved: 2026-01-23)

Kaya, Z., & Bagçeci, B. (2025). Investigation of the effect of project-based teaching on the distance education process. *Journal of Pedagogical Research and Educational Perspectives*, 4(1), 45–60. https://doi.org/10.29329/pedper.2025.80 (Retrieved: 2026-01-23)

Konin.pl. (2020). *Raport z badania ankietowego na temat zdalnej edukacji w szkołach podstawowych i ponadpodstawowych na terenie miasta Konina*. Konin.pl. https://www.konin.pl/files/dokumenty/szkola_naszych_marzen/zdalna_edukacja_wyniki_raport.pdf (Retrieved: 2026.01.23)

Kvale, S. (2007). *Doing interviews*. Sage Publications. ISBN 9780761949770. https://doi.org/10.4135/9781849208963

Levin, I., Semenov, A. L., & Gorsky, M. (2025). Smart learning in the 21st century: Advancing constructionism across three digital epochs. *Education Sciences, 15*(1), 45. https://doi.org/10.3390/educsci15010045 (Retrieved: 2026-01-23)

Montiel, H., & Gómez-Zermeño, M. G. (2021). Educational challenges for computational thinking in K–12 education: A systematic literature review. *Computers, 10*(6), 69. https://doi.org/10.3390/computers10060069 (Retrieved: 2026-01-23)

Ormazábal, I., Borotto, F. A., & Astudillo, H. F. (2021). An agent-based model for teaching–learning processes. *Physica A: Statistical Mechanics and its Applications, 565*, 125563. https://doi.org/10.1016/j.physa.2020.125563 (Retrieved: 2026-01-23)

Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism: Research reports and essays* (pp. 1–11). Ablex Publishing.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. Basic Books. ISBN 0-465-01063-6 (Retrieved: 2026-01-23)

Punie, Y. (Ed.), & Redecker, C. (2017). *European framework for the digital competence of educators: DigCompEdu* (EUR 28775 EN). Publications Office of the European Union. https://doi.org/10.2760/178382 (Retrieved: 2026-01-23)

Railsback, S. F., & Grimm, V. (2019). *Agent-based and individual-based modeling: A practical introduction* (2nd ed.). Princeton University Press. ISBN 978-0-691-19082-2

Resnick, M. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT Press. ISBN 9780262344340

Saab, N., van Joolingen, W., & van Hout-Wolters, B. (2011). Support of the collaborative inquiry learning process. *Metacognition and Learning, 7*(1), 7–23. https://doi.org/10.1007/s11409-011-9068-6 (Retrieved: 2026-01-23)

Skaraki, E., & Kolokotronis, F. (2022). Preschool and early primary school age children learning of computational thinking through the use of asynchronous learning environments in the age of Covid-19. *Advances in Mobile Learning Educational Research, 2*(1), 180–186. https://doi.org/10.25082/AMLER.2022.01.002 (Retrieved: 2026-01-23)

Wen, C.-T., Liu, C.-C., Chang, H.-Y., Chang, C.-J., Chang, M.-H., Fan Chiang, S.-H., Yang, C.-W., & Hwang, F.-K. (2020). Students' guided inquiry with simulation and its relation to school science achievement and scientific literacy. *Computers & Education, 149*, 103830. https://doi.org/10.1016/j.compedu.2020.103830 (Retrieved: 2026-01-23)

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215 (Retrieved: 2026-01-23)

Yin, R. K. (2014). *Case study research: Design and methods* (5th ed.). Sage Publications. ISBN 9781452242569

Zhang, F. (2025). Where inquiry-based science learning meets gamification. *International Journal of Educational Technology in Higher Education*. Advance online publication. https://doi.org/10.1080/0144929X.2024.2433058 (Retrieved: 2026-01-23)

Appendix A.

Interview questions covering the main thematic areas

1. Motivations and the context of the project's creation

- What inspired you to create a simulator for the competition?
- How did you perceive the significance of the pandemic theme in the context of your own life and learning?
- What emotions accompanied you while working on the project during the period of social isolation?

2. Work process and creative development

- How did you plan the project – from the initial idea to its implementation?
- What sources did you use (materials, consultations, videos, online examples)?
- Which tools and elements of Scratch did you use most frequently?
- How did you cope with technical problems and bugs in the code?

3. Difficulties and problem-solving strategies

- What was the most challenging aspect of creating the simulator?
- What strategies did you use to overcome these challenges?
- Did anyone assist you in solving problems (teacher, peers, online forums)?

4. Analysis of design decisions (process tracing)

- Why did you choose a self-service store as the environment for the simulation?
- How did you decide which parameters (e.g., number of customers, masks, distancing) were most important?
- How did you test the effectiveness of the simulator?
- What changes or improvements did you introduce in successive iterations of the project?
- What led you to select the final version submitted to the competition?

5. Reflections and significance of the project

- What did you learn through creating this project – both technically and personally?
- In your view, how did the simulator affect other students or users?
- How do you assess the significance of this project for your own development today?

Appendix B.

Key Program Modules

1. Customer Movement Module

Each customer is an independent object (sprite) moving randomly within the store space (Fig. 1). Movement is implemented using a set of control blocks: forever, move X steps, and if on edge, bounce. The direction is periodically randomized – every few cycles, the program selects a new direction (point in direction (pick random 0–360)). Collision-avoidance conditions are included: if another customer is detected within a predefined distance, the agent adjusts its path. This module illustrates computational thinking through the decomposition of the problem into units (agents) and the implementation of algorithms and control loops (Brennan & Resnick, 2012).

2. Infection Module

A portion of the customers is designated as initial virus carriers according to a starting parameter chosen by the player (Fig. 2). Their behaviour includes random events: within a forever loop, the instruction if (random < probability of sneezing) → show virus particles is executed. Each virus particle is a separate sprite that moves for a defined period (repeat N times → move K steps), after which it disappears (hide). The particle's lifetime simulates the limited viability of the virus. Global variables (lifetime, number of infected) are used to update interface indicators. This module demonstrates the application of conditionals and randomness – key components of computational thinking – and a simplified representation of probabilistic processes.

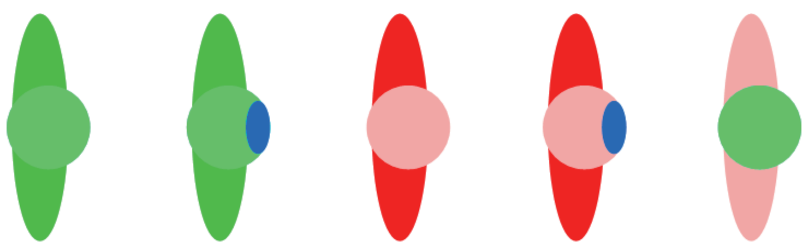

Figure 2. Types of agents (customers) in the simulation.

Note: **1** – healthy without a mask; **2** – healthy with a mask (added by the player); **3** – infected without a mask; **4** – infected with a mask (added by the player); **5** – healthy individual infected during the simulation.

Source: Own work.

3. Interaction and Infection Module

Infection occurs when a virus particle touches another agent without a mask. The module uses a collision-based condition (if touching [customer] and [mask = false] → set [infected = true]). The probability of infection is modified by mask usage (if mask = true → random < 10%, if mask = false → random < 40%). This module combines conditional event control with probabilistic reasoning and randomness. Interactions between agents form a simplified multi-agent model that illustrates principles of emergent modelling, in which global patterns arise from local rules (Railsback & Grimm, 2019).

4. User Panel and Results Module

The user panel contains variables and counters (number of infected individuals, number of healthy individuals, percentage of mask usage). Users may manipulate parameters through buttons or input variables (change number of customers, add masks). The simulation outcome (infection rate) is updated in real time and displayed at the top of the screen. Visual elements (colours, messages, animations) and auditory cues (recorded sneezing and coughing samples) reinforce the didactic effect and enhance user engagement with causal observation.

Figure 3. Overview of a fragment of the simulation code.

Source: Own work.

5. Logic and Dependencies in the Code

The entire program is built on the interaction of infinite loops and conditional statements, creating a dynamic system responsive to randomness and agent interactions (Fig. 3). The hierarchy of events unfolds as follows:

1. Starting the game triggers the creation of agents (when green flag clicked).
2. Each agent acts independently within a forever loop.
3. Unmasked infected agents generate virus particles at set intervals.
4. Virus particles check for collisions with unmasked customers and update their infection status.
5. A global variable tallies the total number of infections.
6. Results are presented to the user in real time.

The mechanics combine elements of determinism (movement modelling), randomness (sneezing events), and interaction (encounters within the store), enabling the user to observe correlations between simulation settings and outcomes (Wen et al., 2020).

Maria Wisniewska, Zbigniew Wisniewski

Symulacja rozprzestrzeniania się wirusa oparta na języku Scratch jako konstruktywistyczny projekt e-learningowy

S t r e s z c z e n i e

Artykuł przedstawia jakościowe studium przypadku projektu symulacyjnego stworzonego w środowisku Scratch przez uczennicę szkoły podstawowej w okresie nauczania zdalnego. Celem badania jest analiza, w jaki sposób projektowanie prostego modelu agentowego w wizualnym środowisku programowania może wspierać rozwój kompetencji cyfrowych i komputacyjnych. Analiza opiera się na trzech źródłach danych: artefakcie stworzonym w Scratch, dokumentacji konkursowej oraz retrospektywnym wywiadzie półustrukturyzowanym z autorką projektu.

Wyniki wskazują, że konstruowanie symulacji sprzyjało rozwijaniu kluczowych praktyk myślenia komputacyjnego – takich jak dekompozycja, iteracyjne udoskonalanie oraz rozwiązywanie problemów – a także umożliwiło uczennicy eksplorację zależności przyczynowo-skutkowych w ramach uproszczonego modelu. Projekt pokazuje, że dostępne środowiska programowania mogą skutecznie wspierać uczenie się konstrukcjonistyczne, angażując uczniów w projektowanie i testowanie działających artefaktów. Wyniki sugerują również, że projekty oparte na symulacjach mogą wzbogacać edukację zdalną i hybrydową, oferując możliwości aktywnego eksperymentowania i refleksyjnej analizy.

Głównym ograniczeniem badania jest jego jednostkowy charakter, który zawęża możliwość uogólniania wniosków. Wymagane są dalsze badania porównawcze i długofalowe, aby ocenić szerszy wpływ edukacyjny symulacji tworzonych przez uczniów.

S ł o w a k l u c z o w e: konstrukcjonizm, Scratch, nauczanie zdalne, myślenie komputacyjne, kompetencje cyfrowe, kompetencje obywatelskie

A Scratch-Based Simulation of Virus Spread as a Constructionist E-Learning Project

Maria Wisniewska, Zbigniew Wisniewski

Una simulación de la propagación de un virus en Scratch como proyecto e-learning construccionista

Resumen

Este artículo presenta un estudio de caso cualitativo sobre un proyecto de simulación desarrollado en Scratch por una estudiante de educación primaria durante el periodo de aprendizaje remoto. El objetivo del estudio es analizar cómo el diseño de un modelo basado en agentes en un entorno de programación visual puede favorecer el desarrollo de competencias digitales y de pensamiento computacional. El análisis se basa en tres fuentes de información: el artefacto creado en Scratch, la documentación del concurso y una entrevista retrospectiva semiestructurada realizada a la autora del proyecto.

Los resultados muestran que la construcción de la simulación permitió a la estudiante desarrollar prácticas clave del pensamiento computacional – como la descomposición, la mejora iterativa y la resolución de problemas – y explorar relaciones causales dentro de un modelo simplificado. El proyecto demuestra cómo los entornos de programación accesibles pueden facilitar el aprendizaje construccionista al involucrar al alumnado en el diseño y prueba de artefactos ejecutables. Asimismo, los resultados sugieren que los proyectos basados en simulaciones pueden enriquecer el aprendizaje remoto e híbrido, ofreciendo oportunidades para la experimentación activa y el análisis reflexivo.

La principal limitación del estudio es su naturaleza de caso único, lo que restringe la posibilidad de generalizar las conclusiones. Se requieren estudios comparativos y longitudinales adicionales para evaluar el impacto educativo más amplio de las simulaciones creadas por estudiantes.

Palabras clave: construccionismo, Scratch, enseñanza a distancia, pensamiento computacional, competencias digitales, competencias cívicas

Мария Вишневска, Збигнев Вишневски

Симуляция распространения вируса в среде Scratch как конструкционистский e-learning-проект

Аннотация

В статье представлен качественный кейс-стади проекта симуляции, созданного в среде Scratch ученицей начальной школы в период дистанционного обучения. Цель исследования – проанализировать, каким образом разработка простейшей агентной модели в визуальной программной среде может способствовать развитию цифровых компетенций и вычислительного мышления. Анализ опирается на три источника данных: созданный в Scratch артефакт, документацию конкурса и ретроспективное полуструктурированное интервью с автором проекта.

Полученные результаты показывают, что процесс конструирования симуляции помог учащейся развить ключевые практики вычислительного мышления – такие как декомпозиция, итеративное совершенствование и решение проблем – а также исследовать причинно-следственные связи внутри упрощённой модели. Проект демонстрирует, что доступные визуальные программные среды могут эффективно поддерживать конструкционистский подход, вовлекая учащихся в создание и тестирование работоспособных артефактов. Кроме того, результаты

свидетельствуют о том, что симуляционные проекты могут обогащать дистанционное и гибридное обучение, предоставляя возможности для активного экспериментирования и рефлексивного анализа.

Основным ограничением исследования является его характер единичного кейса, что сужает возможности обобщения выводов. Для оценки более широкого образовательного эффекта студенческих симуляций необходимы дальнейшие сравнительные и лонгитюдные исследования.

Ключевые слова: конструкционизм, Scratch, дистанционное обучение, вычислительное мышление, цифровые компетенции, гражданские компетенции