



From Imagination to Digital Action: Developing a Contextual Scratch-Based Coding Model for Generation Alpha in Primary Education

Sri Untari¹, Agung Hariyono², Nindyawati Nindyawati³, Rafli Syahputra⁴, Muhammad Fachrif

^{1,2,3,4,5} Universitas Negeri Malang, Jl. Semarang No. 5, Malang 65145, Jawa Timur, Indonesia

sri.untari.fis@um.ac.id

Abstract. *This study aims to develop and describe a contextual Scratch-based coding learning model that is relevant to the characteristics of Generation Alpha in elementary schools. The research employs a Research and Development (R&D) approach using the Hannafin and Peck model, which consists of needs analysis, design, and development and implementation stages. A mixed-methods approach was also applied through observations, interviews, documentation, and questionnaires. The research subjects included fourth- and fifth-grade students and teachers at the Laboratory Elementary School of Universitas Negeri Malang in Blitar. The findings indicate that the Scratch-based coding learning model enhances student engagement, digital creativity, and computational thinking skills. "The developed model was proven valid (87.5%), practical (88%), and effective, as indicated by a significant improvement in student learning outcomes (pre-test = 62.4, post-test = 84.6, N-gain = 0.59, $p < 0.05$). These results confirm that Scratch-based coding learning is an adaptive pedagogical approach that not only improves technical skills but also strengthens digital creativity and computational thinking as essential competencies for the 21st century.*

Keywords: coding, Scratch, Generation Alpha, computational thinking, project-based learning

1. Introduction

The rapid transformation of education in the 21st century has shifted the focus of learning from mere cognitive achievement toward the development of higher-order competencies, including creativity, critical thinking, collaboration, and digital literacy [1]. This shift is further intensified by the emergence of Generation Alpha—children born after 2010—who are growing up in a highly dynamic digital ecosystem characterized by ubiquitous technology, artificial intelligence, and interactive media environments [2], [3]. As digital natives from early childhood, they demonstrate distinct learning preferences, such as visual orientation, interactivity, instant feedback, and exploratory engagement.

Consequently, traditional teacher-centered approaches are increasingly insufficient to address their learning needs, particularly in primary education contexts. In response to these challenges, coding education has gained global attention as a strategic approach to fostering essential 21st-century skills. Coding is no longer viewed solely as a technical skill but as a cognitive and creative process that supports problem-solving, logical reasoning, and innovation [4], [5]. In particular, computational thinking encompassing decomposition, pattern recognition, abstraction, and algorithmic thinking—has been widely recognized as a foundational competence for future citizens in the digital era [6]. Integrating coding into early education, therefore, is not only relevant but necessary to equip learners with the skills required to navigate and shape a technology-driven world.

Among various coding platforms, Scratch has emerged as one of the most widely adopted tools for introducing programming concepts to children. Developed by the MIT Media Lab, Scratch utilizes a block-based visual programming interface that allows learners to create animations, games, and interactive stories without requiring advanced technical knowledge [7], [8]. This visual approach lowers the cognitive barrier to entry and enables students to focus on logic, creativity, and problem-solving rather than syntax. Previous studies have shown that Scratch-based learning can enhance students' engagement, motivation, and computational thinking skills, while also promoting creativity and collaborative learning experiences [9], [10].

Despite its potential, the integration of coding in elementary education remains limited and fragmented in many contexts. In practice, coding is often treated as an extracurricular activity rather than an integral component of the formal curriculum. Several challenges persist, including limited teacher competence in coding pedagogy, lack of structured instructional models, and insufficient alignment with students' developmental characteristics [11]. Moreover, existing studies tend to focus on the effectiveness of coding tools rather than on the design of

pedagogically grounded learning models that are contextually relevant and adaptable to specific learner characteristics, particularly those of Generation Alpha.

This gap highlights the need for a systematic and pedagogically grounded approach to designing coding learning environments in primary education. Instructional design models such as the Hannafin and Peck framework offer a structured yet flexible approach, emphasizing iterative processes of needs analysis, design, and development to ensure alignment between learner characteristics, learning objectives, and instructional strategies [12]. When combined with constructivist principles project-based learning approaches, such models have the potential to create meaningful, student-centered learning experiences that promote active knowledge construction and creativity. However, there remains a lack of empirical research that integrates Scratch-based coding, project-based learning, and systematic instructional design frameworks into a coherent pedagogical model specifically tailored for Generation Alpha in elementary school settings. This indicates a clear research gap not only in terms of instructional design but also in contextual adaptation and pedagogical innovation.

Therefore, this study aims to develop and describe a Scratch-based coding learning model that is contextually relevant and pedagogically aligned with the characteristics of Generation Alpha in elementary education. Specifically, the study explores how the model is designed and implemented, how it supports student engagement and learning experiences, and how it impacts students' digital creativity and computational thinking skills. By integrating the Hannafin and Peck instructional design model with project-based learning and visual programming environments, this study contributes to the development of an adaptive pedagogical framework for digital learning in primary education. The novelty of this research lies in the integration of three key dimensions: (1) a systematic instructional design approach based on the Hannafin and Peck model, (2) the use of Scratch as a visual programming environment to support creative digital production, and (3) the alignment of learning strategies with the cognitive and behavioral characteristics of Generation Alpha. This integrated model not only addresses the limitations of existing practices but also offers a scalable and contextually relevant framework for enhancing digital learning innovation in elementary schools.

2. Research Methods

2.1 Research Design

This study employed a Research and Development (R&D) approach aimed at developing and validating a Scratch-based coding learning model for elementary school students. The study adopted the Hannafin and Peck instructional design model, which consists of three iterative phases: needs analysis, design, and development and implementation [12]. This model was selected due to its flexibility in accommodating continuous evaluation and refinement, making it particularly suitable for developing technology-integrated learning models. To comprehensively capture both the development process and its impact, this study also utilized a mixed-methods approach. The integration of qualitative and quantitative data enabled a more holistic understanding of the learning model, including its practicality, effectiveness, and user experience. The mixed-methods design in this study followed a convergent approach, in which qualitative and quantitative data were collected and analyzed concurrently, then integrated to provide a comprehensive interpretation of the findings.

2.2 Research Context and Participants

The study was conducted at the Laboratory Elementary School of Universitas Negeri Malang in Blitar, Indonesia. This setting was selected due to its readiness in terms of digital infrastructure and its openness to pedagogical innovation. The participants consisted of elementary school students from Grades IV and V, as well as teachers who were directly involved in the implementation of the Scratch-based coding learning model. A total of 30 students participated in the implementation phase, while teachers contributed to both the needs analysis and evaluation stages. The selection of participants was based on purposive sampling, considering their relevance to the objectives of the study, particularly in relation to early exposure to digital learning environments.

2.3 Data Collection Techniques and Instruments

Data were collected using multiple techniques to ensure methodological triangulation. These included observation, interviews, documentation, and questionnaires. Observation was conducted to examine student engagement, participation, and interaction during the learning process. Semi-structured interviews were carried out with teachers to gain insights into instructional challenges, perceptions, and the practicality of the developed

model. Documentation, including student project outputs and learning artifacts, was analyzed to assess creativity and the application of computational thinking.

In addition, questionnaires were administered to measure students' learning interest and responses to the Scratch-based learning activities. The questionnaire employed a Likert scale to capture students' perceptions quantitatively. Furthermore, a computational thinking test and a project assessment rubric were used to evaluate students' learning outcomes, focusing on indicators such as problem decomposition, pattern recognition, abstraction, and algorithmic thinking.

2.4 Data Analysis Techniques

Qualitative data obtained from observations, interviews, and documentation were analyzed using thematic analysis. This process involved data reduction, coding, categorization, and interpretation to identify emerging patterns related to student engagement, creativity, and learning experiences. Quantitative data were analyzed using descriptive and inferential statistics. Descriptive analysis was used to present the overall trends in student responses and learning outcomes, while inferential analysis was conducted using the Wilcoxon signed-rank test to determine the significance of differences between pre-test and post-test scores. The use of the Wilcoxon test was appropriate given the small sample size and non-parametric nature of the data. To measure the effectiveness of the learning model, the normalized gain score (N-gain) was calculated to determine the extent of improvement in students' learning outcomes. The interpretation of gain scores followed established criteria, categorizing improvements into low, medium, and high levels.

2.5 Validity and Trustworthiness

The validity of the developed learning model was assessed through expert judgment involving subject matter experts and instructional design experts. The validation process focused on three main aspects: content relevance, instructional design, and media usability. The practicality of the model was evaluated based on teacher responses during the implementation phase, while its effectiveness was measured through improvements in student learning outcomes and engagement levels. To ensure the trustworthiness of qualitative data, this study applied triangulation by comparing findings from multiple data sources and collection methods. In addition, member checking was conducted with teachers to validate the interpretation of interview data. These strategies enhanced the credibility and reliability of the research findings.

3. Result and Discussion

3.1 Generation Alpha and Digital Learning Transformation

Generation Alpha represents a cohort of learners who are inherently immersed in digital environments from an early age. Unlike previous generations, their interaction with technology is not supplementary but embedded within their daily experiences, shaping how they access information, communicate, and learn [3], [13]. As a result, they tend to exhibit preferences for visual, interactive, and exploratory learning environments that provide immediate feedback and opportunities for active engagement. This shift necessitates a transformation in pedagogical approaches, particularly in primary education. Traditional instructional models that rely heavily on passive knowledge transmission are increasingly misaligned with the cognitive and behavioral characteristics of Generation Alpha. Contemporary educational discourse emphasizes the need for learner-centered, technology-integrated approaches that support active participation and meaningful learning experiences [14]. In this context, digital learning environments are not merely tools for content delivery but spaces for interaction, creation, and knowledge construction. However, despite the growing recognition of these needs, many educational practices still struggle to effectively align instructional strategies with the characteristics of Generation Alpha. This indicates a gap between the evolving nature of learners and the pedagogical models implemented in classrooms, highlighting the urgency of developing adaptive and contextually relevant learning designs.

3.2 Scratch-Based Coding and Computational Thinking in Primary Education

Coding education has been widely acknowledged as a powerful means of fostering computational thinking and problem-solving skills among young learners. Computational Thinking, as introduced by Jeannette Wing, encompasses a set of cognitive processes including decomposition, pattern recognition, abstraction, and

algorithmic thinking, which are essential for addressing complex problems in the digital age [15] Among the various tools available, Scratch has become one of the most prominent platforms for introducing coding to children. Its block-based visual interface allows learners to construct programs by manipulating graphical elements, reducing the complexity associated with traditional text-based programming languages [16]. This design not only lowers barriers to entry but also encourages experimentation, creativity, and iterative learning.

Empirical studies have demonstrated that Scratch-based learning environments can significantly enhance student engagement, motivation, and learning outcomes. In addition, Scratch supports the development of digital creativity by enabling learners to produce interactive artifacts such as animations, games, and stories [17]. These activities align with the principles of constructionism, which emphasize learning through the creation of meaningful products.

Nevertheless, existing research often focuses on the effectiveness of Scratch as a tool rather than examining how it can be systematically integrated into structured pedagogical models. As a result, the implementation of Scratch in classrooms frequently lacks coherence and consistency, limiting its potential impact on deeper learning processes such as computational thinking and creativity.

3.3 Instructional Design, Project-Based Learning, and Digital Pedagogy

The integration of coding into primary education requires not only appropriate tools but also well-designed instructional frameworks. Instructional design models provide systematic approaches to aligning learning objectives, content, activities, and assessment. One such model, the Hannafin and Peck framework, emphasizes iterative processes consisting of needs analysis, design, and development, ensuring that instructional solutions are responsive to learner needs and contextual conditions [12] This approach is closely aligned with constructivist learning theories proposed by Jean Piaget and Lev Vygotsky, which highlight the importance of active engagement, social interaction, and experiential learning in knowledge construction. Within this paradigm, learners are not passive recipients of information but active participants who construct meaning through exploration and collaboration.

In addition, Project-Based Learning (PjBL) has been widely recognized as an effective pedagogical approach for fostering meaningful learning experiences. By engaging students in authentic tasks and problem-solving activities, PjBL promotes deeper understanding, creativity, and collaboration [18]. When combined with digital tools such as Scratch, PjBL creates opportunities for learners to design and develop digital artifacts, thereby integrating technical skills with creative expression. Despite these theoretical advancements, there remains a lack of integrative models that combine instructional design frameworks, coding tools, and learner characteristics into a cohesive pedagogical approach. Most existing studies address these components in isolation, resulting in fragmented practices that do not fully leverage the potential of digital learning environments.

3.4 Research Gap and Conceptual Direction

Based on the reviewed literature, it is evident that three key dimensions—Generation Alpha characteristics, Scratch-based coding, and instructional design frameworks—have been widely discussed but rarely integrated into a unified pedagogical model. Existing research tends to emphasize either the technological aspect (e.g., coding tools) or the cognitive outcomes (e.g., computational thinking), with limited attention to the design of learning processes that holistically connect these elements. This gap highlights the need for a structured and contextually grounded learning model that integrates visual programming environments, project-based learning, and systematic instructional design. Such a model is essential to ensure that coding education in primary schools is not only engaging but also pedagogically meaningful and aligned with learners' developmental characteristics.

Therefore, this study proposes a Scratch-based coding learning model developed through the Hannafin and Peck framework, designed to support the learning needs of Generation Alpha. This model aims to bridge the gap between theory and practice by providing a coherent structure that integrates digital tools, pedagogical strategies, and learner-centered design principles.

3.5 Needs Analysis

The needs analysis reveals that coding instruction in primary schools remains at an early stage of implementation, despite the availability of basic digital infrastructure. Initial observations showed that 70% of students had never been exposed to block-based coding. The presence of computer laboratories and digital devices indicates institutional readiness; however, their pedagogical utilization is not yet optimal. This gap suggests that technological provision alone is insufficient without corresponding instructional design and teacher readiness.

This finding aligns with prior studies emphasizing that the successful integration of coding in primary education depends not merely on infrastructure but on teachers' pedagogical capacity and curriculum alignment [19]. Furthermore, the flexible use of students' personal devices reflects an emergent form of adaptive digital learning, commonly found in developing educational contexts. This condition illustrates a shift toward blended learning environments, where formal and informal digital practices intersect.

From a theoretical perspective, this stage highlights the importance of contextualizing computational thinking education within real classroom constraints. It also reinforces the argument that digital transformation in education requires systemic alignment between infrastructure, pedagogy, and learning culture.

3.6 Instructional Design

The instructional design phase focuses on structuring learning activities that integrate digital literacy with computational thinking skills. The learning sequence progresses from basic logic introduction to project-based application, allowing students to gradually construct their understanding. This design reflects constructivist principles, particularly those proposed by Piaget and Vygotsky, where learners actively construct knowledge through interaction and experience. The staged progression also aligns with the computational thinking framework introduced by [20], which emphasizes decomposition, pattern recognition, abstraction, and algorithmic thinking [21]. Moreover, the integration of project-based learning within the design strengthens its pedagogical relevance. By embedding coding tasks within meaningful projects, students are not only exposed to technical concepts but also engaged in authentic problem-solving processes. This supports the notion that effective digital learning environments must bridge conceptual understanding with practical application.

Thus, the instructional design developed in this study demonstrates a synthesis between constructivist learning theory and contemporary computational thinking frameworks, making it pedagogically robust for primary education contexts.

3.7 Implementation and Student Engagement

The implementation phase shows a significant increase in student engagement during coding activities. Students actively participated in developing digital projects such as animations and simple games, demonstrating high levels of motivation and curiosity. This finding can be interpreted through the lens of constructionism, as proposed by Papert (1980), which posits that learning becomes more meaningful when learners actively create tangible artifacts. The Scratch environment enables such engagement by providing a visual and intuitive platform where abstract programming concepts are transformed into concrete outputs.

Additionally, the observed engagement supports previous empirical findings that project-based coding activities enhance students' participation and problem-solving abilities [22]. The interactive and exploratory nature of Scratch aligns with the learning preferences of Generation Alpha, who are characterized by their affinity for visual and digital environments. Importantly, this increased engagement is not merely behavioral but also cognitive, as students demonstrate deeper involvement in planning, testing, and revising their digital projects. This suggests that coding activities can foster sustained engagement that integrates both affective and cognitive dimensions of learning.

3.8 Development of Digital Creativity

One of the most significant findings of this study is the enhancement of students' digital creativity. The diversity of student-generated projects indicates high levels of originality, variation, and expressive capacity. This outcome supports the concept of digital creativity, where technology functions as a medium for creative expression rather than merely a tool for consumption [23]. Students were able to experiment with various Scratch features, such as loops, sounds, and interactive elements, to produce unique digital artifacts. For instance, students successfully created digital products such as educational maze games, folklore animations, and interactive quizzes using Scratch.

Furthermore, this finding aligns with [24], who argues that creativity develops when learners are given opportunities to explore, experiment, and reconstruct digital elements. The project-based structure of the learning model provides such opportunities, enabling students to engage in divergent thinking processes. From a broader perspective, the results also resonate with the Technological Pedagogical Content Knowledge (TPACK) framework [25], which emphasizes the integration of technology and pedagogy to support creative learning. In

this study, Scratch serves as a pedagogical tool that facilitates the intersection of technological skills and creative expression.

Thus, the development of digital creativity observed in this study reflects not only improved technical competence but also a transformation in how students interact with technology—as active creators rather than passive users.

3.9 Improvement of Computational Thinking Skills

The findings indicate a notable improvement in students' computational thinking abilities, particularly in decomposition, pattern recognition, abstraction, and algorithmic thinking. During the coding process, students demonstrated the ability to break down complex tasks into smaller steps, identify patterns in program behavior, and construct logical sequences using block-based commands. These skills are fundamental components of computational thinking as defined by [20]

This result is consistent with the work of [15], who highlight that early exposure to coding activities can significantly enhance higher-order thinking skills. The Scratch environment supports this development by providing a low-barrier entry point to programming concepts while still allowing for complex problem-solving. Importantly, the improvement in computational thinking extends beyond technical skills. It also contributes to general problem-solving abilities, suggesting that coding education can play a broader role in cognitive development. This reinforces the argument that computational thinking is a core competency for the 21st century, applicable across various domains. [16]

3.10 Model Validation: Validity, Practicality, and Effectiveness

The developed learning model demonstrates high levels of validity, practicality, and effectiveness. Expert validation results indicate a validity score of 87.5%, suggesting that the model meets both conceptual and instructional standards. The practicality of the model is reflected in teachers' responses, with an average score of 88%. Teachers reported that the learning sequence is clear, manageable within classroom time constraints, and aligned with students' characteristics. This highlights the feasibility of implementing the model in real classroom settings. In terms of effectiveness, the model shows a significant improvement in student learning outcomes.

The increase from a pre-test average of 62.4 to a post-test average of 84.6, with a gain score of 0.59, indicates a moderate to high level of learning improvement. Additionally, observational data show that 93% of students were actively engaged during the learning process. These findings align with the principles of educational design research, which emphasize that a successful learning model must be valid, practical, and effective [26]. The results suggest that the developed model not only meets theoretical expectations but also demonstrates strong empirical performance. Overall, the validation results confirm that the Scratch-based coding learning model is a viable pedagogical innovation that supports digital learning transformation in primary education.

4 Conclusions

This study demonstrates that the Scratch-based coding learning model is effective in enhancing students' learning engagement, digital creativity, and computational thinking skills at the primary school level. The visual and interactive nature of Scratch facilitates meaningful learning experiences that align with the characteristics of Generation Alpha. Project-based learning provides opportunities for students to develop ideas creatively through diverse digital products. In this context, coding is positioned not merely as a technical skill, but as a pedagogical medium that fosters exploration, expression, and problem-solving.

From a development perspective, the model has been proven to be valid, practical, and effective, indicating its potential as an adaptive pedagogical framework for supporting digital learning transformation in primary education. The findings highlight that coding education should be framed as a transformative approach that integrates imagination, creativity, and computational thinking within digital learning ecosystems. Overall, the Scratch-based coding learning model demonstrates strong potential as an innovative pedagogical approach that transforms students' imagination into creative, reflective, and meaningful digital action, while also contributing to the advancement of adaptive digital pedagogy in primary education. This study provides empirical evidence for the integration of coding, instructional design, and learner characteristics in a unified pedagogical model.

5 Suggestions

The findings of this study offer important implications for primary education practices, particularly in integrating coding as part of digital literacy. The developed model suggests that project-based approaches combined with visual programming can serve as effective strategies to address the learning needs of Generation Alpha. Additionally, the results underscore the importance of strengthening teachers' competencies in digital pedagogy. Without adequate pedagogical understanding, coding instruction risks being reduced to purely technical activities, rather than fostering students' creativity and critical thinking skills.

Based on the findings, several recommendations can be proposed. First, Scratch-based coding learning should be integrated into the primary school curriculum as part of digital literacy and 21st-century skills development. This integration can be implemented through thematic learning, as well as co-curricular and extracurricular activities. Second, teachers should receive continuous professional development in coding pedagogy, computational thinking, and project-based digital learning design. Strengthening teacher capacity is essential to ensure that Scratch is utilized not only as a technical tool but also as a meaningful pedagogical medium.

Third, schools and educational policymakers are encouraged to provide inclusive digital infrastructure, including devices, internet access, and child-friendly digital learning resources. Policy support is also needed to promote adaptive and innovative coding practices aligned with the characteristics of Generation Alpha. Finally, future research is recommended to test the model in broader educational contexts, across different grade levels, and by incorporating additional variables such as digital literacy, digital citizenship, and long-term impacts on students' critical and creative thinking skills.

References

- [1] M. C. Martínez-Bravo, C. Sádaba Chalezquer, and J. Serrano-Puche, "Dimensions of Digital Literacy in the 21st Century Competency Frameworks," *Sustainability*, vol. 14, no. 3, p. 1867, Feb. 2022, doi: 10.3390/su14031867.
- [2] M. McCRINDLE and A. Fell, "To Ruth, with whom this adventure of life is a great joy and who, along with our children Acacia, Jasper, Zari, Brighton and Corban, brings immeasurable flourishing."
- [3] J. M. Twenge, "More Time on Technology, Less Happiness? Associations between Digital-Media Use and Psychological Well-Being," *Curr. Dir. Psychol. Sci.*, vol. 28, no. 4, pp. 372–379, Aug. 2019, doi: 10.1177/0963721419838244.
- [4] P. Tuomi, J. Multisilta, P. Saarikoski, and J. Suominen, "Coding skills as a success factor for a society," *Educ. Inf. Technol.*, vol. 23, no. 1, pp. 419–434, Jan. 2018, doi: 10.1007/s10639-017-9611-4.
- [5] Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., ... & Stupurienė, G. (2022), *Reviewing computational thinking in compulsory education: state of play and practices from computing education*. LU: Publications Office, 2022. doi: 10.2760/126955.
- [6] H. Belmar, "Review on the teaching of programming and computational thinking in the world," *Front. Comput. Sci.*, vol. 4, p. 997222, Oct. 2022, doi: 10.3389/fcomp.2022.997222.
- [7] B. Batni, S. N. Junaini, J. Sidi, W. A. Mustafa, and Z. I. Awang Ismail, "Current Research Trends of Scratch Block based Programming for K-12: A Systematic Review," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 51, no. 2, pp. 138–152, Sep. 2024, doi: 10.37934/araset.51.2.138152.
- [8] Y. Hu, C.-H. Chen, and C.-Y. Su, "Exploring the Effectiveness and Moderators of Block-Based Visual Programming on Student Learning: A Meta-Analysis," *J. Educ. Comput. Res.*, vol. 58, no. 8, pp. 1467–1493, Jan. 2021, doi: 10.1177/0735633120945935.
- [9] W. P. L. Tarigan, P. Paidi, A. Wiyarsi, S. Suhartini, and S. Fajri, "Integrating Scratch with Project-Based Learning to Cultivate Students' Academic Performance," *J. Innov. Educ. Cult. Res.*, vol. 5, no. 4, pp. 723–735, Dec. 2024, doi: 10.46843/jiecr.v5i4.1994.
- [10] A. O. Sunday, F. J. Agbo, J. Suhonen, I. Jormanainen, and M. Tukiainen, "Co-designing to develop computational thinking skills in Nigeria K-12 using scratch," *Educ. Inf. Technol.*, vol. 30, no. 11, pp. 14925–14965, Jul. 2025, doi: 10.1007/s10639-025-13386-y.
- [11] M. N. Giannakos, I. O. Pappas, L. Jaccheri, and D. G. Sampson, "Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness," *Educ. Inf. Technol.*, vol. 22, no. 5, pp. 2365–2382, Sep. 2017, doi: 10.1007/s10639-016-9538-1.
- [12] M. J. Hannafin, and K. L. Peck., "The Design, Development, and Evaluation of Instructional Software. Macmillan Publishing Company,," 1987.
- [13] M. McCRINDLE and A. Fell, "To Ruth, with whom this adventure of life is a great joy and who, along with our children Acacia, Jasper, Zari, Brighton and Corban, brings immeasurable flourishing."
- [14] S. Goswami, "A Student-centric Approach to Higher Education in an Era of Technology-assisted Learning – How Far Active Learning is Possible?."

- [15] S. Grover and R. Pea, "Computational Thinking in K–12: A Review of the State of the Field," *Educ. Res.*, vol. 42, no. 1, pp. 38–43, Jan. 2013, doi: 10.3102/0013189X12463051.
- [16] M. Resnick *et al.*, "Scratch: programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009, doi: 10.1145/1592761.1592779.
- [17] S. Papavlasopoulou, K. Sharma, and M. N. Giannakos, "Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences," *Comput. Hum. Behav.*, vol. 105, p. 105939, Apr. 2020, doi: 10.1016/j.chb.2019.03.003.
- [18] K. A. Maspul, "Enhancing Project-Based Learning in STEM Education with Integrated Technology and Coding," *J. Intell. Syst. Inf. Technol.*, vol. 1, no. 1, pp. 16–24, Jan. 2024, doi: 10.61971/jisit.v1i1.20.
- [19] K. A. Mills, J. Cope, L. Scholes, and L. Rowe, "Coding and Computational Thinking Across the Curriculum: A Review of Educational Outcomes," *Rev. Educ. Res.*, vol. 95, no. 3, pp. 581–618, Jun. 2025, doi: 10.3102/00346543241241327.
- [20] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006, doi: 10.1145/1118178.1118215.
- [21] J. Clarke-Midura, V. R. Lee, J. F. Shumway, D. Silvis, J. S. Kozlowski, and R. Peterson, "Designing formative assessments of early childhood computational thinking," *Early Child. Res. Q.*, vol. 65, pp. 68–80, 34 2023, doi: 10.1016/j.ecresq.2023.05.013.
- [22] A. Yadav, H. Hong, and C. Stephenson, "Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms," *TechTrends*, vol. 60, no. 6, pp. 565–568, Nov. 2016, doi: 10.1007/s11528-016-0087-7.
- [23] M. Resnick, "Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play".
- [24] N. Behnamnia, A. Kamsin, M. A. B. Ismail, and A. Hayati, "The effective components of creativity in digital game-based learning among young children: A case study," *Child. Youth Serv. Rev.*, vol. 116, p. 105227, Sep. 2020, doi: 10.1016/j.childyouth.2020.105227.
- [25] D. Maor, "Using TPACK to develop digital pedagogues: a higher education experience," *J. Comput. Educ.*, vol. 4, no. 1, pp. 71–86, Mar. 2017, doi: 10.1007/s40692-016-0055-4.
- [26] J. Krajcik, K. L. McNeill, and B. J. Reiser, "Learning-goals-driven design model: Developing curriculum materials that align with national standards and incorporate project-based pedagogy," *Sci. Educ.*, vol. 92, no. 1, pp. 1–32, Jan. 2008, doi: 10.1002/sce.20240.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

