Technology, Knowledge and Learning
https://doi.org/10.1007/s10758-022-09601-1

ORIGINAL RESEARCH

The WearableLearning Platform: A Computational Thinking Tool Supporting Game Design and Active Play

Ivon Arroyo[1] · Avery Harrison Closser[2] · Francisco Castro[1] · Hannah Smith[2] · Erin Ottmar[2] · Matthew Micciolo[2]

Accepted: 10 March 2022

Abstract

This emerging technology report introduces the *WearableLearning* (WL) platform as a tool to exercise computational thinking and STEM learning for 5-12th grade students through mobile technology-augmented active game play and game creation. Freely available at *WearableLearning.org,* it allows students and teachers to play, create, debug, and manage multiplayer, active games. To date, *WearableLearning* has been used in schools and after-school programs by roughly 500 students and 25 teachers to create games covering STEM curricular content. *WearableLearning* enables the creation of physically active and social games, while offering possibilities for research on computational thinking, embodied cognition, collaborative learning, game-based learning, and practical applications of technology in STEM classrooms.

Keywords Computational thinking · Educational games · Mobile technology-supported learning

1 Introduction and Description of the Emerging Technology

Computational Thinking (CT)—the cognitive processes that are required to formulate problems and reach a solution that may be executed by a computer (Wing, 2014)—has been widely recognized by educators, researchers, and business leaders as a set of skills that are increasingly necessary to improve access to STEM careers (Smith, 2016; Weintrop et al., 2016; Adams & Pruim, 2012; Wing, 2014). In this paper, we introduce *WearableLearning*, a technology platform that enables STEM teachers and their 5-12th grade students to play, create, debug, and manage multiplayer mobile technology-based games, without the need of any prior programming experience (Arroyo et al., 2017; Micciolo, 2018), as a way to develop STEM content knowledge and CT.

WearableLearning provides students with opportunities to develop an understanding of CT concepts by visually programming games modeled as Finite State Machines (FSMs), a

✉ Ivon Arroyo
ivon@cs.umass.edu

[1] University of Massachusetts Amherst (Computer Science), Amherst, MA, USA

[2] Worcester Polytechnic Institute (Social Science & Policy Studies), Worcester, MA, USA

topic typically reserved for computer science undergraduate students (Hopcroft & Ullman, 1979). Distinct from other block-based and visual programming environments for children (Maloney et al., 2010; Kafai & Burke, 2014), *WearableLearning* enables the creation and deployment of collaborative team-based games that support active, embodied experiences, thus also acting as a testbed for further investigation of the importance of embodiment and games in learning (Daily et al., 2015; Sung et al., 2017).

WearableLearning supports collaborative, interactive, and gamified experiences for full classes of K-12 students and teachers, where the ecology of the classroom embraces mobile technologies that are familiar to kids. The technology guides and supports students as they manipulate objects available in their physical environments (e.g., to take measurements, make computations, estimate variables, or gesture using their bodies), all while interacting with their peers, teachers, and technology, in rich discourse. As players, students participate in games that involve problems from relevant curriculum, reinforcing subject knowledge. As creators, students and teachers engage with CT practices and concepts, by considering how to design engaging games for their peers, how to blend STEM material into games based on their own knowledge, and how to specify game mechanics thoroughly by describing and programming the behavior of the devices as Finite State Machines (see Fig. 3).

The programming environment of *WearableLearning* requires no prior programming experience. Unlike other programming languages for children (e.g. Scratch: Maloney et al., 2010), apps for mobile devices (Kodable: 2021; Tynker, 2021), or those that use Finite State Machines (e.g. Kara: Hartmann et al., 2001), *WearableLearning*'s game-editing platform enables the creation of *multi-player* game experiences, allowing game creators to program different behaviors for players and teams, and define alternative parallel cell phone behaviors for the different players in the game (if this is desired). This functionality adds the idea of a game-wide, team-wide, or player-based "scope", and encourages game designers to consider situations of two or more players participating in events or circumstances that can happen or co-exist at the same time. In Computer Science terms, this feature corresponds to analyzing problems of *concurrency*, an advanced concept for young programmers, albeit in very accessible terms[1]. In this case, the concurrency problem is distributed between human players, objects in the environment, and the mobile devices (see Fig. 2 and Fig. 4). As with any programming language, programmers can debug their games by launching a simulator of a cell phone, allowing for iterative refinement. In *WearableLearning's* debugger, programmers have to choose which team/player to debug, as there could be alternative programs created for different players in a game.

From the *WearableLearning* website, students and teachers can access the platform in any of the following three capacities: Game Editors, Game Players, or Game Managers.

1.1 Game Managers

WearableLearning allows teachers and students to facilitate game play as Game Managers. Game Managers can browse the *WearableLearning* platform interface to find open-access student-, teacher-, or researcher- created games to play with students as class-wide activities, complete with instructions and materials. Game Managers can select and start games

[1] For instance, in some forms of "scavenger hunt", it might not be desirable that all players look for the same object at the same time, but different ones.

from any device with WiFi accessibility, generating a PIN for players to join the specified game.

1.2 Game Players

WearableLearning allows students to play games using any mobile device with Internet access (Fig. 2). Students may wear, instead of carry, their mobile devices, which frees up their hands to manipulate their environment and handle game materials, or use their full bodies (e.g., run, climb) without hassle (Howison et al., 2011; Johnson et al., 2013). Students enter a PIN provided by the game manager and select a role (i.e., team and player) to join the game.

1.3 Game Editors

The *WearableLearning* game-editing environment is designed for programming novices, so that younger students can create unique games with a wide variety of content, levels, and modes of collaborative or competitive play, for players or teams to engage in. As Game Editors, students define games as *Finite State Machines* (FSMs) that describe the behavior of players' mobile devices (see Fig. 1). FSMs are computational models that can be used to simulate phenomena, sequential logic, and the behavior of computer programs. They include a finite number of states and directional transitions between states (including loops). The *WearableLearning* curriculum guides students from the design to the implementation and deployment processes of their games (Castro et al., under review).

Game Editors program their games using *WearableLearning's* drag-and-drop, FSM-based programming language, where they specify *output states* to indicate media to display on mobile screens (e.g., text, images, videos) for the purpose of providing game instructions, questions, or hints. The states can be configured to display different media to make interesting prompts such as displaying a math problem with pictures related to the problem, combined with an audio recording. Game Editors also specify *transitions* between states to indicate a possible input that players can submit (e.g., button presses, text input, color sequences) to move from one state to the next (see Figs. 1 and 2a.). The transition input specification enables different game response interactions; for example, Game Editors can ask players to respond to problems with a text entry (e.g., "How many sides does a triangle have?), or use buttons to enter choices (e.g., "Push the black button if you want a hint"). Game Editors can also test-run and debug their games in the built-in debugger, which simulates game play on a phone simulator, providing immediate feedback (Fig. 2b.).

2 Relevance for Learning, Instruction, and Assessment

Computational Thinking at the K-12 level is a relatively new concept, yet it includes much more than just coding (Weintrop et al., 2016). Some CT concepts and practices involve problem decomposition (e.g., breaking problems down into sub-problems), identifying patterns of behaviors, representing different forms of data (forms of inputs, object identifiers, outputs, etc.), thinking at different levels of abstraction (stepping back to understand a full picture of a solution, black-boxing details as a way to simplify and manage complexity), specifying algorithms, evaluating solutions, and iterative refinement (Grover & Pea, 2018).

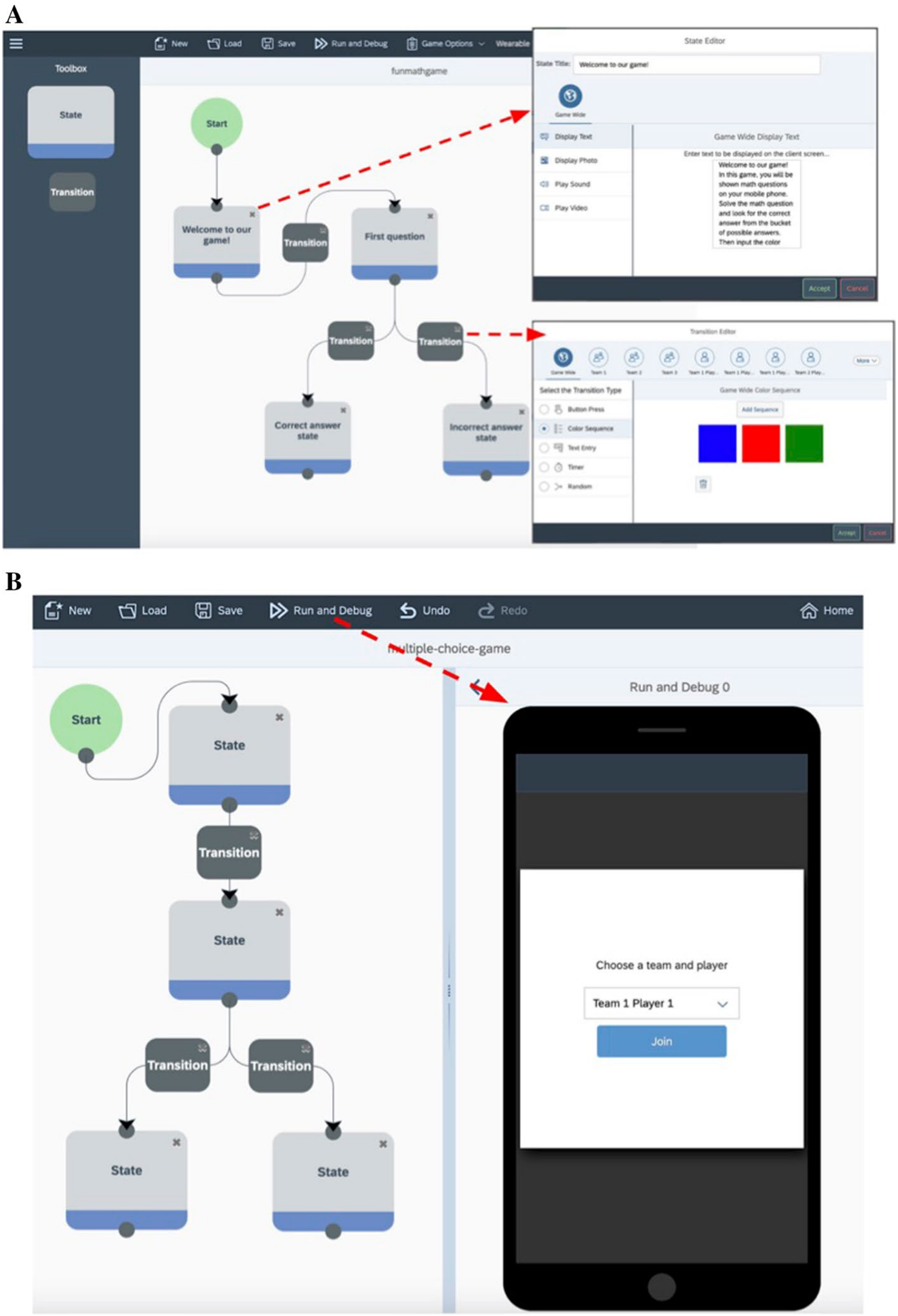

Fig. 1 **a** *WearableLearning Game Editor*. Left: Each box corresponds to an output state, and each arrow is associated to an input transition event. Top right: State editor. Bottom right: Input Transition editor. **b** Clicking "Run and Debug" allows Game Editors to debug the game within a simulator; the debugger steps through each state of the game

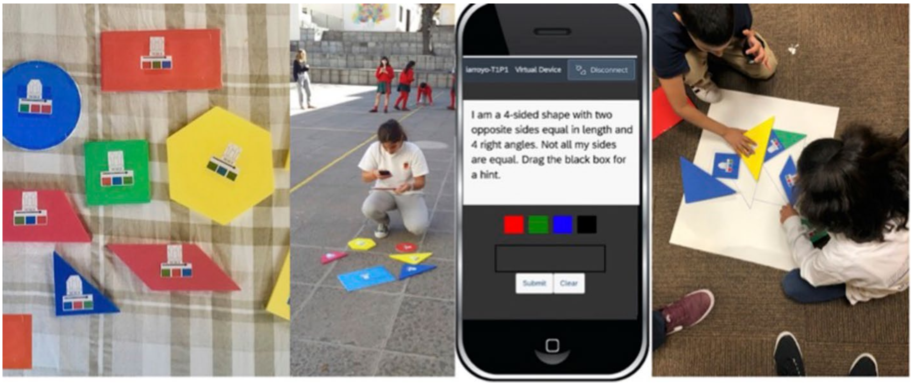

Fig. 2 6th–7th grade students playing a sample game during *Stage 1* of the *WearableLearning* curriculum

Students can learn these skills as they create and play games such as the ones that *WearableLearning* supports.

All students should have the opportunity to learn and practice CT in school. However, until recently, CS education has not typically been offered to students until high school and many schools do not even offer it, especially in schools of high minority populations (Margolis, 2008). Students should have the opportunity to practice CT in K-12 schooling because such skills are facets of higher-level thinking and problem solving that are applicable across contexts, even beyond Computer Science. However, many children do not have adequate opportunities to learn or practice these skills, partially because few teachers have the tools, training, and curricula required to teach them (Howard, 2019; Schulte et al., 2012; Tsai & Chai, 2012; Yildiz Durak, 2020). *WearableLearning* provides opportunities for K-12 students and teachers to develop CT skills through a low-barrier entry point for gameplay and game design, both through the platform and curriculum.

We have outlined a CT framework that maps each stage of the game play and creation curriculum to CT concepts and practices (Castro et al., under review). This framework describes how students exercise CT concepts by engaging with the *WearableLearning* platform and curriculum. This framework will guide future research on the extent in which *WearableLearning* promotes CT abilities among students.

3 Emerging Technology in Practice

We have created the *WearableLearning* curriculum to help students and teachers move through a multi-stage game creation process by first designing their games on paper, then specifying FSMs, then implementing these in the editor (Fig. 3), and last playing one another's games (Fig. 4). Both the *WearableLearning* curriculum and the platform have been successfully deployed with roughly 500 upper elementary (5-6th grade), middle (7th-8th grade), and high school (9-12th grade) students and roughly 25 math and STEM teachers. While games could be created regarding any domain, our team has conducted studies to investigate whether and how math games created with *WearableLearning* can be integrated into STEM classrooms to promote mathematics learning (Arroyo et al., 2021; Casano, Agapito, Arroyo & Rodrigo, 2019; Agbaji, 2019).

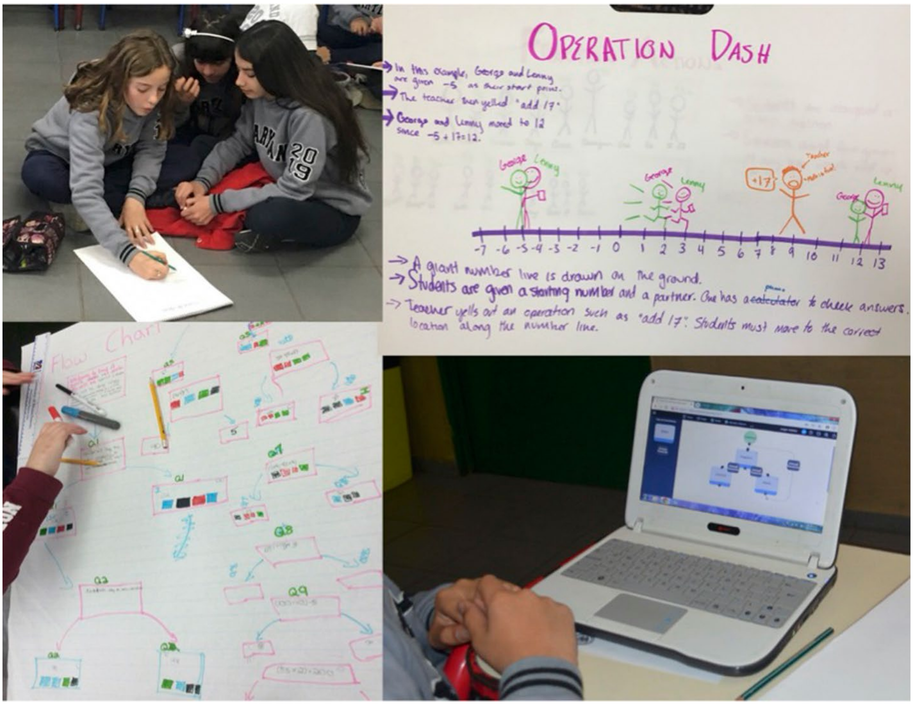

Fig. 3 The process of creating games in *WearableLearning*. Upper Left: Stage 2: 6th Grade students brainstorming games on paper pads, using any representation they want. Upper Right: A product of Stage 2, by a team of 10th graders; Bottom Left: Stage 3: 5th grade students specifying their finite state machine diagrams in paper pads. Bottom Right: Stage 4: 6th grade students transferring their FSMs from paper to the Game Editor

Fully-fledged games have been created by teachers, students, and researchers using *WearableLearning*. For instance, the teacher-created game "Integer Hopscotch" is a movement-based game designed for students to practice working with positive and negative integers. A 20-foot number line with integers is laid out on the floor for a team of students to use. Students work together to solve word problems as they move along the number line (see more at *wearablelearning.org/#games*). Students input each answer into a phone to receive immediate feedback and can request a hint about whether they should move forward or backwards. The goal of the game is to work in pairs to complete all of the integer problems by "walking along the number line". Forty-nine (49) students played *IntegerHopscotch* and showed significant gains from pretest to posttest after playing the game in one class period (Arroyo et al., 2021; Smith et al., 2020). Beyond learning from the games, teachers have found that their students also enjoy playing the physically active mobile games built in *WearableLearning*.

Regarding game-creation by students, one hundred and fifty-three (153) 6th-7th grade students in three schools in Argentina completed the play, design, create, play cycle of the curriculum during seven weeks (one hour per week) on a Spanish-translated version of *WearableLearning*. Figure 4 shows a group of these students playing a game created by another group of students on the last day. Figure 3 shows a class of 5th-grade students in a public school in Massachusetts, as well as their game products,

Fig. 4 *WearableLearning* curriculum Stage 5: 6th grade children playing the games their classmates have created. Note how they incorporate elements from the environment: ropes, hoola-hoops, candy, baskets/containers, in addition to the phones

during different stages of the curriculum, including the brainstorming stage, finite-state-machine design stage, and programming stage.

Our research so far has focused on assessing the feasibility for students to create physically-active multiplayer math games in *WearableLearning*, and identifying any challenges that they might encounter during the game creation process. We have found much evidence that high school, middle, and even upper elementary school students can successfully create games with *WearableLearning* (Arroyo et al., 2017; Harrison et al., 2018). We have also designed and conducted a seven-hour teacher professional development program to help train math and STEM teachers to implement *WearableLearning* in their own classrooms (Smith et al., 2020). In this program, we provided support to teachers as they played and created their own games, tested the effectiveness of their games with their own students, then supported their own students in the game creation process using *WearableLearning*. Teachers and students alike reported positive feelings about the usability of *WearableLearning* as a whole (Smith et al., 2020).

4 Conclusions and Significant Challenges

WearableLearning provides a platform that allows teachers and students to exercise CT, a collection of processes at the intersection of engineering design, problem solving, and iterative design processes. In *WearableLearning*, the final product is a game that involves multiple players and their environment, much like playground games. Hidden beneath those game products, are design processes and computational thinking. Students engage in this cyclical process by playing and iteratively designing games in *WearableLearning*. For instance, students and teachers must anticipate, articulate, specify, program, implement, test, and revise at a high level of detail all possible moves that a game player could take within each game. Additionally, they must simultaneously consider the interactions between multiple players, the STEM concepts the game is about, physical materials, as well as spaces in the environment.

Our current challenges are two-fold. First, one challenge is also a motivation behind this project: many K-12 teachers are often not comfortable with using technology or teaching CT in the classroom. Prior research has indicated barriers that teachers face in integrating technology into their classrooms (Ertmer, 1999; Tsai, & Chai, 2012). To break these barriers, teachers need more opportunities to experience working with technology and design processes to appreciate the challenges and thinking that are required to develop effective games for student learning. As such, we need to train teachers to implement technologies to positively shift their confidence in the importance, value, and relevance of CT in their students' future success. This process involves teacher buy-in, time to learn, create, and implement, as well as time for students to learn and apply the computational and design processes put forth by the technology later.

Another challenge is assessing how students and teachers develop CT skills in the gameplay and design process, as there is not a general agreement on the boundaries of CT, nor how to assess it broadly (only partial assessments of CT skills now exist, e.g., Román-González, 2015). Recently, researchers have shifted to defining and measuring CT within the context of specific projects. To that end, we have created a framework of what, when, and how CT occurs within the *WearableLearning* curriculum (Castro et al., under review).

As we improve our measurement and assessment of CT concepts and practices (e.g., we have now implemented logging of users' actions as they create and play games), we plan to analyze the behaviors of students as they program and play games to better understand how engaging with *WearableLearning* can exercise as well as promote CT in both students and teachers and support their problem-solving and design processes.

Funding This material is based upon work supported by the National Science Foundation under Grants No. IIS-2041785 and IIS-2026722. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Data availability Not applicable.

Declarations

Conflict of Interests There are no conflicts of interest.

References

Agbaji, E. (2019). *Improving math learning with embodied game-based mobile and wearable technologies* (Unpublished master's thesis). Worcester Polytechnic Institute.

Adams, J. C., & Pruim, R. J. (2012). Computing for STEM majors: enhancing non CS majors' computing skills. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 457–462).

Arroyo, I., Micciollo, M., Casano, J., Ottmar, E., Hulse, T., & Rodrigo, M. (2017). Wearable learning: Multiplayer embodied games for math. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY '17). ACM, pp. 205–216.

Arroyo, I., Castro, F., Smith, H., Harrison Closser, A., Ottmar, E. (2021) Augmenting Embodied Mathematics Classrooms with Mobile Tutors. Proceedings of the American Educational Research Association Annual Meeting.

Castro, F., Closser, A. H., Smith, G., Smith, H., Rasul, I., Arroyo, I., & Perez, L. (under review). Defining a computational thinking framework within the WearableLearning platform and curriculum. Manuscript under review.

Daily, S. B., Leonard, A. E., Jörg, S., Babu, S., Gundersen, K., & Parmar, D. (2015). Embodying computational thinking: Initial design of an emerging technological learning tool. *Technology, Knowledge and Learning, 20*(1), 79–84.

Ertmer, P. A. (1999). Addressing first-and second-order barriers to change: Strategies for technology integration. *Educational Technology Research and Development, 47*(4), 47–61.

Grover, S., & Pea, R. (2018). Computational Thinking: A Competency Whose Time Has Come. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer Science Education: Perspectives on Teaching and Learning in School* (pp. 19–38). Bloomsbury Academic.

Harrison, A., Hulse, T., Manzo, D., Micciolo, M., Ottmar, E., & Arroyo, I. (2018). Computational thinking through game creation in STEM classrooms. *Proceedings (Part II) of the 19th International Conference on Artificial Intelligence in Education*. pp. 134–138.

Hartmann, W., Nievergelt, J., & Reichert, R. (2001). Kara, finite state machines, and the case for programming as part of general education. In: *Proceedings IEEE Symposia on Human-Centric Computing Languages and Environments*, 135–141. https://doi.org/10.1109/HCC.2001.995251

Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley. ISBN 978-0-201-02988-8.

Howard, N. R. (2019). EdTech Leaders' Beliefs: How are K-5 teachers supported with the integration of computer science in K-5 classrooms? *Technology, Knowledge, and Learning, 24*, 203–217. https://doi.org/10.1007/s10758-018-9371-2

Howison, M., Trninic, D., Reinholz, D., & Abrahamson, D. (2011). The Mathematical Imagery Trainer: from embodied interaction to conceptual learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1989–1998).

Johnson, K., Pavleas, J., & Chang, J. (2013). Kinecting to mathematics through embodied interactions. *Computer, 46*(10), 101–104.

Kafai, Y., & Burke, Q. (2014). *Connected Code: Why Children Need to Learn Programming*. The MIT Press. https://doi.org/10.2307/j.ctt9qf8rk

Kodable (2021) http://www.kodable.com/. Last accessed January 2022.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, *10*(4), 6. https://doi.org/10.1145/1868358.1868363

Margolis, J. (2008). *Stuck in the Shallow End: Education*. MIT Press.

Micciolo, M. (2018). *Designing a visual programming language for the creation of multiplayer embodied games* (Unpublished master's thesis). Worcester Polytechnic Institute.

Román-González, Marcos. (2015). Computational Thinking Test: Design Guidelines and Content Validation. Proceedings of the EDULEARN15 Conference. https://doi.org/10.13140/RG.2.1.4203.4329

Schulte, C., Hornung, M., Sentance, S., Dagiene, V., Jevsikova, T., Thota, N., ... & Peters, A. K. (2012). Computer science at school/CS teacher education: Koli working-group report on CS at school. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research* (pp. 29–38).

Smith, M. (2016). Computer Science for All. Archives of the White House of President Barak Obama. https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all Retrieved March 23, 2020.

Smith, H., Closser, A. H., Ottmar, E., & Arroyo, I. (2020). Developing math knowledge and computational thinking through game play and design: A professional development program. *Contemporary Issues in Technology and Teacher Education, 20*(4), 660–686.

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning, 22*(3), 443–463.

Tsai, C. C., & Chai, C. S. (2012). The "third"-order barrier for technology-integration instruction: Implications for teacher education. *Australasian Journal of Educational Technology*, *28*, 1057–1060. https://doi.org/10.14742/ajet.810.

Tynker (2021) https://www.tynker.com/ Last accessed January 2022.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilenski, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*, 127–147.

Wing, J. (2014). Computational Thinking Benefits Society. *Social Issues in Computing*. http://socialissues.cs.toronto.edu. Retrieved February 4, 2022.

Yildiz Durak, H. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology, Knowledge, and Learning, 25*, 179–195. https://doi.org/10.1007/s10758-018-9391-y