Important Note: This is a preprint manuscript. The final paper was published as:
Kirwan, C. Costello, E., & Donlon, E. (2018*) Computational Thinking and Online Learning: A Systematic Literature Review*. In Proceedings of the: 17th European Conference on eLearning, ECEL 2018, Athens, Greece, November 1-2; 2018 (pp 657-650). Academic Conferences and Publishing International Limited, Reading, UK.

Computational Thinking and Online Learning: A Systematic Literature Review

Colette Kirwan, Eamon Costello, Enda Donlon
Dublin City University, Ireland
Colette.Kirwan@dcu.ie
Eamon.Costello@dcu.ie
Enda.Donlon@dcu.ie

Abstract: This paper introduces research concerned with investigating how Computational Thinking and online learning can be successfully married to help empower secondary teachers to teach this subject. To aid this research, a systematic literature review was undertaken to investigate what is currently known in the academic literature on where Computational Thinking and online learning intersect. This paper presents the findings of this systematic literature review. It outlines the methodology used and presents the current data available in the literature on how Computational Thinking is taught online. Using a systematic process eight hundred articles were initially identified and then subsequently narrowed down to forty papers. These papers were analysed to answer the following two questions: 1. What are the current pedagogical approaches to teaching Computational Thinking online? 2. What were the categories of online learning observed in the teaching of Computational Thinking? Our findings show that a wide range of pedagogical approaches are used to teach Computational Thinking online, with the constructivist theory of learning being the most popular. The tools used to teach Computational Thinking were also varied, video game design, playing video games, competitions, and unplugged activities, to name a few. A significant finding was the dependency between the tool used and the definition of the term Computational Thinking. Computational Thinking lacks consensus on a definition, and thus the definition stated in the literature changed depending on the tool. By considering a significant body of research up to the present, our findings contribute to teachers, researchers and policy makers understanding of how computational thinking may be taught online at second level.

Keywords: Computational Thinking, Online Learning, Pedagogy, Secondary Education7

1. Introduction

There are two challenges to integrating Computational Thinking into the classroom 1) the lack of consensus regarding a definition and 2) the shortage of qualified teachers that can teach this skill (Menekse 2015; Curzon et al. 2014). This lack of agreement was highlighted in 2009 when a workshop organised by the US National Research Council with the goal of establishing "The Scope and Nature of Computational Thinking" failed to reach consensus among its participants concerning the content and structure of Computational Thinking (National Research Council (NRC) 2010, p.65). Computational Thinking has its beginning with Seymour Papert (1980) and his much citied book "Mindstorms: Children, Computers, and Powerful Ideas" but it was Jeannette Wing, who popularised Computational Thinking more widely. She defined it as follows: "Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (Wing 2006).

This definition of Wing's has significance for compulsory education in that it states that Computational Thinking is fundamentally a thought process, i.e. independent from technology and that its solutions can be executed by either a human or computer, or both (Bocconi et al. 2016, p.15). Notwithstanding Wing's definition, other subsequent definitions and views have arisen such as Aho (2011), Barr and Stephenson (2011), Royal Society definition (Fuber 2012), Computing At Schools Barefoot Organisation (2014), Bocconi et al. (2016) and Shute et al. (2017). What is interesting about these definitions, is that there is no explicit mention of programming languages. Wing's 2006 paper was very specific about this stating that Computational Thinking was about conceptualisation, not programming. It was concerned with ideas, not artefacts. This is not a universal belief. At the aforementioned workshop organised by the US National Research Council in 2009 Roy Pea, Ursula Wolz, Mitchel Resnick and Eric Roberts all voiced opinions concurring that programming is essential to Computational Thinking (NRC 2010, pg 13). Ioannidou et al., (2011) provide a definition that relates to Computational Thinking Patterns and Scalable Game Design. Alternatively, Brennan and Resnick (2012) proposed a definition of Computational Thinking that revolved around the Scratch language. The above discussion highlights that no simple answer is forth-coming for the question what is Computational Thinking, thus illustrating one of the challenges to integrating Computational Thinking into the classroom.

The second challenge to integrating Computational Thinking in the classroom is the shortage of qualified teachers that can teach this subject (Menekse 2015; Curzon et al. 2014). An online course or platform may aid in helping overcome this challenge. This online course/platform may be used to support the teaching and learning of Computational Thinking inside and outside the classroom, self-directed or teacher lead. This research proposes investigating what is known in the literature about online courses and platforms in relation to the teaching and learning of Computational Thinking. Note, online was defined based on the Online Learning Consortium (OLC) definitions (Sener 2015) (web-based courses were included if part of the teaching was performed online or if the tool was self-contained, i.e. could be used without the presence of a teacher).

2. Methodology: Systematic Literature Review

A systematic literature review was undertaken with the initial aim of investigating what is currently known in the academic literature on the intersection of Computational Thinking, online learning and secondary students. This research is underpinned by a pragmatic worldview (specifically the teachings of John Dewey), and thus a systematic approach to the identification of the literature was taken with the goal of providing a transparent and accountable methodology that can be replicated (Gough et al. 2012, pg 18).

The PICO (Population, Intervention, Comparison and Outcome) framework was used to guide the specification of the research questions and the inclusion strategy for the review. With respect to Secondary School students: **Q1** What are the current pedagogical approaches to teaching Computational Thinking online? **Q2** What were the categories of online learning observed in the teaching of Computational Thinking? The intention was to ensure that the returned literature was concerned with all three topics: Computational Thinking, online learning and secondary school students (Gough 2012 chapter 4, pg 103). The PICO elements used were as follows:

Population: K12 children age 12-18
Intervention: Teaching of Computational Thinking online without using a programming language
Comparison: Teaching of Computational Thinking online using a programming language
Outcome: Effectiveness and extensiveness of the teaching medium and the effectiveness of the pedagogical approach

A search strategy was developed with the purpose of choosing search terms that would strike a balance between sensitivity (finding all articles on a subject) and specificity (finding relevant articles) (EPPI Centre 2006). The PICO framework provided the keywords (“K12 students,” “Computational Thinking” “Online” “Teaching”) that were used to generate synonyms that were Boolean ORed together. For example, the term algorithmic thinking and logical thinking were used as synonyms for Computational Thinking, as they can be considered a subset of Computational Thinking (Syslo (2015) cited in Buitrago Flórez et al., 2017). The final search string was constructed by ANDing the OR lists of synonyms for “Computational Thinking”, “teaching”, and “online”, which was then run against six databases (Inspec, Compendex, ACM Digital Library, Education Research Complete, ERIC and British Education Index.) The K12 synonyms were removed to broaden the search criteria and return more literature. In total 800 articles were returned and subjected to the study selection process.

2.1 Study Selection Process

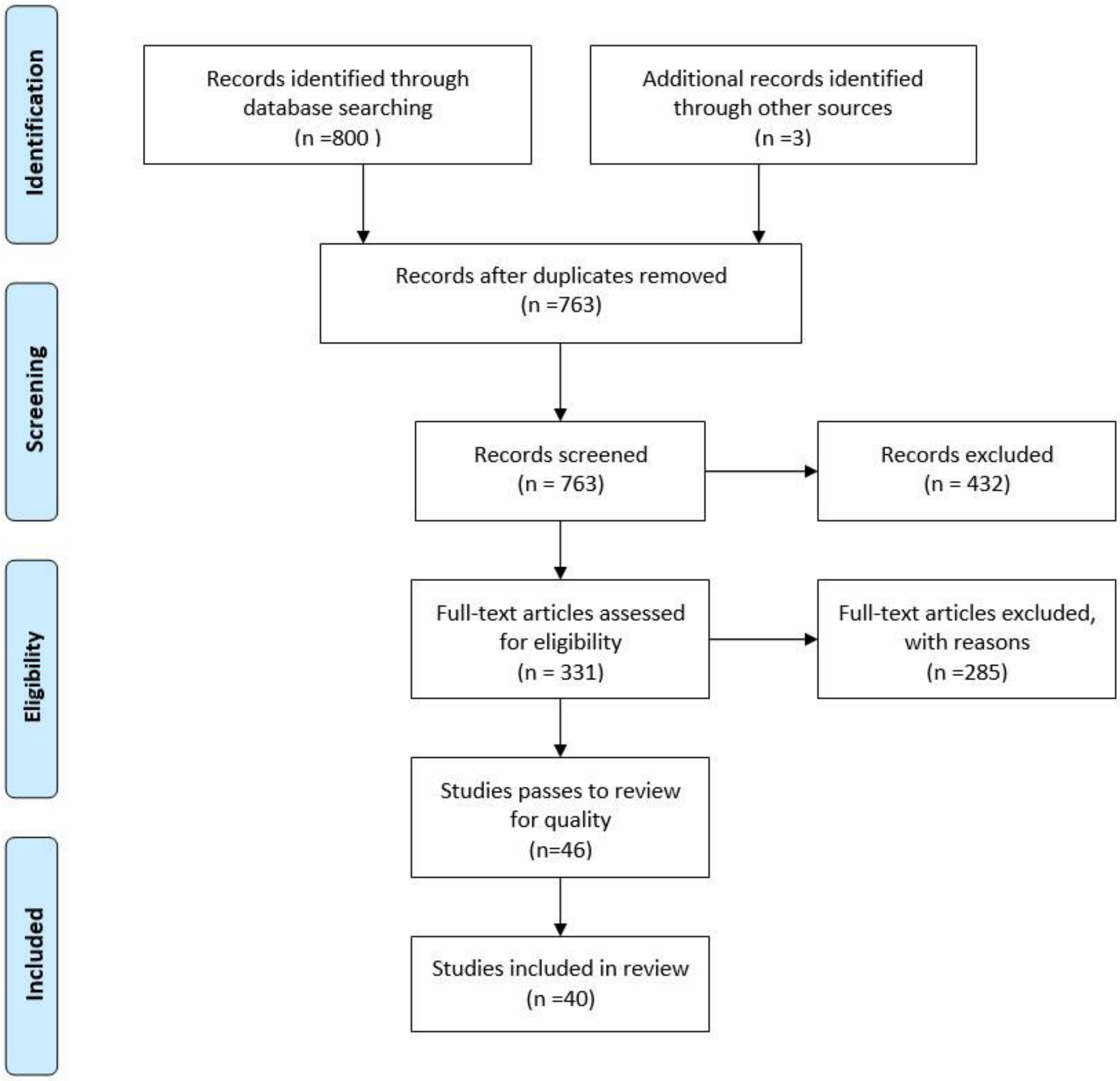

Figure 1 Study Selection Process. Diagram Adapted from Prisma Group (Moher et al. 2009)

As recommended by Kitchenham and Charters (2007), inclusion and exclusion criteria were developed, with the abstract and title of each of the 800 articles being screened against these criteria. Full details of the selection process are depicted in Figure 1. The final result saw 46 articles being eligible for the literature review, and thus screened for quality against the five prompts provided by Dixon-Woods et al. (2006).

The papers were analysed to answer the following two questions:
1. What are the current pedagogical approaches to teaching Computational Thinking online?
2. What were the categories of online learning observed in the teaching of Computational Thinking?

This paper deals specfically with the answering of the first research question. To aid in this task, the reviewed literature was grouped into categories related to the teaching tool type. These categories were Visual programming languages, Playing Video Games, Programming Languages (excluding visual), Unplugged Activities, Competitions, Collaboration Tool and Other.
To facilitate comparison, each category is presented and discussed using the same structure, 1) tools 2) theories and techniques used in the teaching of Computational Thinking for example (collaboration, remixing etc.) and 3) notable points. Literature outside the reviewed forty papers was included in the discussion when required to aid in the explanation of the aforementioned content.

2.2 Findings

2.3 Visual Programming Language (Game Design)

Visual Programming Languages were the most popular approach observed in the literature, appearing in seventeen papers. The languages used in the teaching of Computational Thinking in these papers can be essentially broken down into two categories, Scratch and Agent modelling, i.e. AgentSheet derivatives. AgentSheets is a drag and drop programming language that facilitates the creation of agent based games and simulations (AgentSheets, 2014).

Constructionism is the pedagogical approach most usually associated with visual programming languages especially as the learning is concerned with building an artefact, for example, a game (Ahmadi et al. 2012; Marcelino et al. 2017). This too was found to be borne out in the selected game design literature, with constructionism being the most common approach used (Marcelino et al. 2017; Pellas and Peroutseas 2016). An explicit dependency between this pedagogical approach and the aforementioned tools (Scratch and AgentSheets) was expected but what was unexpected was the dependency that existed between the pedagogical approach, the teaching tool and the definition of Computational Thinking. This dependency was found to be present in the majority of game design papers, thus highlighting the transferability and utility of how Computational Thinking is taught The AgentSheet research drew on a definition of Computational Thinking that incorporated the use of Computational Thinking Patterns (Marshall 2011; Ahmadi et al. 2012; Ahmadi and Jazayeri 2014; Koh et al. 2014; Basawapatna 2016). Computational Thinking Patterns are based on the mechanical phenomena that are relevant to both game design and science simulations, for example, collisions, pushing, pulling etc. (Ioannidou et al.. 2011). This enables students to transfer the knowledge learned from creating abstracted programming patterns in the game design tools to model scientific phenomena (Basawapatna et al. 2011). The majority of the papers that were concerned with the Scratch environment revolved around Brennan and Resnick's (2012) mapping of Scratch blocks to Computational Thinking concepts, for example, sequences, conditionals etc. (Dasgupta et al. 2016; Xie and Abelson 2016; Marcelino et al. 2017). Jenkins (2015) questioned how synonymous Computational Thinking (taught using Scratch) is with problem solving when teachers are assessing it by programming concepts of loops and sequences etc. (Jenkins 2015). Transferability was also a concern of Grover et al. (2015) and was discussed as part of their Foundations for Advanced Computational Thinking (FACT) framework.

While constructionism was the dominant learning theory, it was not the only framework observed. The following frameworks were also adhered to: Zone of Proximal framework (Basawapatna et al. 2013; Koh et al. 2014; Escherle et al. 201), Foundations for Advanced Computational Thinking framework (FACT) and Five Flow of Inspiration Principles (Repenning et al. 2009). The Zone of Proximal Framework is concerned with engagement. The FACT framework is concerned with students learning strong, transferable knowledge and skills (Grover and Pea 2016; Grover et al. 2015). The five flow of inspiration principles (Repenning et al. 2009) was devised and used in the indirect teaching of Computational Thinking using an online homework submission system, which was developed to encourage peer to peer interaction.

Of particular mention in relation to visual programming languages, are the studies which documented how the environment (or process) was the "sole" teaching tool for Computational Thinking. Two studies in this category used this strategy: Ahmadi et al. (2012) and Ahmadi and Jazayeri (2014). The purpose of Ahmadi and Jazayeri (2014) study was to research if a complete novice could learn Computational Thinking skills independently online over a 2.5 hour interval. Using a top-down approach, students were scaffolded to explore the proposed problem first and to gain the programming skills in the course of solving the problem. Their study was successful in meeting this aim, i.e. 90% of tasks were completed in full by students.

Collaboration was an essential pedagogical approach observed in many of the game design studies in this review (Basawapatna and Repenning 2010; Ahmadi and Jazayeri 2014; Marcelino et al. 2017). This approach was greatly aided by the online environment in that it provided both synchronous and asynchronous communication among participants and their teacher. Of particular note are the studies by Repenning et al. (2009) and Basawapatna and Repenning (2010), where they employed an online homework submission system called the Scalable Game Design Arcade (SGDA). The study by Repenning et al. (2009) documents how the Scalable Game Design Arcade was set up to replicate the interactions of a Middle School computer club following the aforementioned Five Flows of Inspiration Principles. The project was created using AgentSheets, but the SGDA provided a supplementary, albeit, indirect way of teaching Computational Thinking using peer to peer interactions. Students learned from each other, by viewing and running each other 's code.
Remixing was another approach used to learn Computational Thinking, and very much in keeping with the constructivist theory of learning, as the constructed is made more meaningful by being placed in a social context (Dasgupta et al. 2016).

2.4 Playing Video Games

The visual programming languages demonstrated how constructing artefacts, for example, a videogame was a vehicle for teaching Computational Thinking. Seven studies demonstrated how simply playing a video game can also be an effective way to practice Computational Thinking, thus enabling students to learn Computational Thinking outside of the classroom. Two of the studies incorporated game design with game playing (Pellas and Peroutseas 2016; Weintrop et al. 2016). Constructionism again was the dominant theory observed. The majority of the games were designed with a "constructionist" activity as core to the game play, for example, Blockyland is a city building game, Make World is concerned with making worlds to teach STEAM concepts. There was only one other theory of note in the papers, that of Computational Encoding (Holbert and Wilensky, 2011), but various techniques such as collaboration remixing, and immersion were also observed (Guenaga et al. 2017; Dhatsuwan and Precharattana 2016; Pellas and Peroutseas 2016). Debabi and Bensebaa (2016) built on Futschek (2006), by proposing the AlgoGame, which allows participants to focus on solving problems rather than learning the syntax of a programming language. They reported that after playing the AlgoGame, an experimental group had better results in writing a selection sort algorithm that a control group.

2.5 Programming language (not visual-based)

Three studies analysed the teaching of computational thinking using tools/languages that do not fall specifically/exclusively into visual based programming language category. Constructivism was once again the most dominant theory observed (Krugel and Hubwieser 2017; Grandell 2005; Berland and Wilensky 2015), but Grandell (2005) course was also influenced by a new conceptual model known as ActiWe (Active on the Web), where the guiding principal is active learning. An interesting didactical dilemma was highlighted by Krugel and Hubwieser (2017) regarding the teaching of Object Oriented Programming. They advocated teaching in a "real life context" but to do that with reference to programming; one has to learn a lot of difficulty concepts quickly. To overcome this difficulty, they initially hid advanced topics, used a strictly object first approach, and ensured fundamentals covered before serious programming covered.

2.6 Un-Plugged Activities

Three articles looked at activities that teach Computational Thinking without using a computer. Two of these articles employed computational creative exercises (Miller et al. 2013; Shell et al. 2014). The goals of these studies were to investigate if the learning of Computational Thinking can be improved if it is combined with creative thinking. These exercises were all hand-ons, and required problems to be solved collaboratively using written analysis and reflection. These computational creative exercises were designed to include the four competencies from Epstein's Generativity Theory on creative thinking, i.e. capturing novelty, challenging accepted norms, broadening knowledge and surrounding oneself with inspiration (Shell et al. 2014).

One study was concerned with teaching Computational Thinking using a mixed approach, i.e. unplugged and programming (Vivian et al. 2014). Programming was introduced after Computational Thinking concepts were understood. Vivian et al. (2014) study was predominantly concerned with the theory behind the development of a Mass Open Online Course for teaching Computer Science. The following online frameworks were discussed: online teacher professional development (oTPD) (structured internet based learning) and technology-mediated professional learning (TMPL). The pedagogical approach taken was a combination of oTPD/extended Massive Open Online Course (xMOOC) and TMPL/connectivism Massive Open Online Course cMOOCs. This enabled them to deliver relevant content knowledge sequentially, but also have a collaborative space where teachers can share knowledge and work together.

2.7 Competitions

Two articles referred to the Bebras competitions (Dagiene and Stupuriene 2016; Izu et al. 2017). The Bebras competition was initially set up to introduce information technology and informatics to students. Its purpose has now changed with it becoming an approach for developing Computational Thinking and for deeper learning of informatics. These competitions can be considered informal learning-by-doing, as students engage in solving both engaging and challenging tasks (Haberman et al., 2011). They can also be used as an innovative way of assessing Computational Thinking (Dagiene and Stupuriene 2016).

2.8 Collaboration Tool

Two particular studies, formed their own category as they were specifically about how the tool is used to develop Computational Thinking using collaboration. The study of Othman et al. (2015) investigated cognitive enhancement in introductory programming through online collaboration. The intervention used a Think-Pairshare approach, i.e. a collective cortex method with the aim of enhancing logical thinking. Wilkerson-Jerde

(2014) study also focused on collaborative environments. Her tool, called the Categorizer, allowed 11-14 year old students to construct, share and categorise computational artifacts that they developed based on fractals.

2.9 Other

Six papers fitted into this last category. Two discussed the teaching of Computational Thinking using the application software Excel (Tsai and Tsai 2017; Tsai et al. 2017) with mixed results. One study focused specifically on what pre-service teachers need to know to teach Computational Thinking outside Computer Science, i.e. in different disciplines, for example, mathematics, science and literacy (Mouza et al. 2017). Liao and Liang's (2017) study focused on learning approaches. Their study sought to research if blended learning can promote computational thinking, and finally, the study by Korkmaz et al. (2017) was concerned with assessment and Computational Thinking Scales. A notable point, is that once again the importance of the Computational Thinking definition was evident. In both Tsai et al. (2017) studies Computational Thinking was defined based on Excel using the definition by Yeh et al. (2011).

3. Conclusion

This paper set out to review the current pedagogical approaches to teaching Computational Thinking online. To facilitate comparison, papers were categorised using their teaching tool. This study identified visual programming languages as the most popular means of teaching Computational Thinking online. This confirms a change in practice; in that, visual programming languages are now being used for students to learn and practice Computational Thinking (Ahmadi and Jazayeri 2014). Constructionism was the most common pedagogical approach observed, a surprising find, was the playing game categories having a "constructionist" activity as core to the game play. Collaboration was also an important approach observed. Two significant finding to emerge from this review are the importance of the Computational Thinking definition, and how it differs depending on the teaching tool, and the relative dearth of research in the teaching of Computational Thinking online using self-directed means

References

AgentSheets, I. (2014) What Is AgentSheets? [online], available: http://www.agentsheets.com/products/index.html.

Ahmadi, N., Jazayeri, M. (2014) "Analyzing the Learning Process in Online Educational Game Design: A Case Study," in *Proceedings of the 2014 23rd Australian Software Engineering Conference*, ASWEC '14, IEEE Computer Society: Washington, DC, USA, 84–93.

Ahmadi, N., Jazayeri, M., Landoni, M. (2012) "Helping Novice Programmers to Bootstrap in the Cloud: Incorporating Support for Computational Thinking into the Game Design Process," in *Proceedings of the 2012 IEEE 12th International Conference on Advanced Learning Technologies*, ICALT '12, IEEE Computer Society: Washington, DC, USA, 349–353.

Aho, A.V. (2011) "Ubiquity symposium: Computation and computational thinking," *Ubiquity*, 2011(January).

Barr, V., Stephenson, C. (2011) "Bringing computational thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?," *ACM Inroads*, 2(1), 48–54.

Basawapatna, A. (2016) "Alexander Meets Michotte: A Simulation Tool Based on Pattern Programming and Phenomenology.," *Journal of Educational Technology & Society*, 19(1), 277–291.

Basawapatna, A., Koh, K.H., Repenning, A., Webb, D.C., Marshall, K.S. (2011) "Recognizing computational thinking patterns," in *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 245–250.

Basawapatna, A.R., Repenning, A. (2010) "Cyberspace Meets Brick and Mortar: An Investigation into How Students Engage in Peer to Peer Feedback Using Both Cyberlearning and Physical Infrastructures," in *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '10, ACM: New York, NY, USA, 184–188.

Basawapatna, A.R., Repenning, A., Koh, K.H., Nickerson, H. (2013) "The zones of proximal flow: guiding students through a space of computational thinking skills and challenges," in *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 67–74. Berland, M., Wilensky, U. (2015) "Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking.," *Journal of Science Education \& Technology*, 24(5), 628–647.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., others (2016) *Developing Computational Thinking in Compulsory Education-Implications for Policy and Practice*.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In AERA2012 - annual meeting of the American Educational Research Association. Vancouver, Canada.

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S. and Danies, G., 2017. Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. Review of Educational Research, 87(4), pp.834-860.

Computing At School Barefoot (2014) Computational Thinking [online], available: http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/. Curzon, P., McOwan, P.W., Plant, N. and Meagher, L.R. (2014). Introducing teachers to computational thinking using unplugged storytelling. IN Proceedings of the 9th Workshop in Primary and Secondary Computing Education (pp. 89-92). ACM.

Dagiene, V., Stupuriene, G. (2016) "Bebras-A Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking," *Informatics in Education*, 15(1). Dasgupta, S., Hale, W., Monroy-Hernández, A., Hill, B.M. (2016) "Remixing As a Pathway to Computational Thinking," in *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work \& Social Computing*, CSCW '16, ACM: New York, NY, USA, 1438–1449.

Debabi, W., Bensebaa, T. (2016) "USING SERIOUS GAME TO ENHANCE ALGORITHMIC LEARNING AND TEACHING.," *Journal of E-Learning \& Knowledge Society*, 12(2), 127–140.

Dhatsuwan, A., Precharattana, M. (2016) "BLOCKYLAND," *Simul. Gaming*, 47(4), 445–464.

Dixon-Woods, M., Cavers, D., Agarwal, S., Annandale, E., Arthur, A., Harvey, J., Hsu, R., Katbamna, S., Olsen, R., Smith, L. and Riley, R., 2006. Conducting a critical interpretive synthesis of the literature on access to healthcare by vulnerable groups. BMC medical research methodology, 6(1), p.35.

EPPI-Centre, 2006. EPPI-Centre methods for conducting systematic reviews. London: EPPI-Centre, Social Science Research Unit, Institute of Education, University of London.

Escherle, N.A., Ramirez-Ramirez, S.I., Basawapatna, A.R., Assaf, D., Repenning, A., Maiello, C., Endo, Y.C., Nolazco-Flores, J.A. (2016) "Piloting Computer Science Education Week in Mexico," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, ACM: New York, NY, USA, 431–436.

Furber, S. (2012) Shut down or Restart? The Way Forward for Computing in UK Schools [online], *The Royal Society, London*, available: https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12computing-in-schools.pdf.

Futschek, G. (2006) "Algorithmic thinking: the key for understanding computer science," in *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*, 159–168.

Gough, D., Oliver, S. and Thomas, J. eds., 2012. An introduction to systematic reviews. Sage.

Grandell, L. (2005) "High School Students Learning University Level Computer Science on the Web - a Case Study of the DASK-Model.," *Journal of Information Technology Education*, 4, 207–218.

Grover, S., Pea, R. (2016) "Designing a blended, middle school computer science course for deeper learning: A design-based research approach," in *Proceedings of International Conference of the Learning Sciences, ICLS*, Singapore, Singapore, 695 – 702.

Grover, S., Pea, R., Cooper, S. (2015) "Designing for deeper learning in a blended computer science course for middle school students.," *Computer Science Education*, 25(2), 199–237.

Guenaga, M., Mentxaka, I., Garaizar, P., Eguiluz, A., Villagrasa, S., Navarro, I. (2017) "Make world, a collaborative platform to develop computational thinking and STEAM," in *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vancouver, BC, Canada, 50 – 59.

Holbert, N.R., Wilensky, U. (2011) "Racing Games for Exploring Kinematics: A Computational Thinking Approach," in *Proceedings of the 7th International Conference on Games + Learning + Society Conference*, GLS'11, ETC Press: Pittsburgh, PA, USA, 109–118.

Ioannidou, A., Bennett, V., Repenning, A., Koh, K.H., Basawapatna, A. (2011) *Computational Thinking Patterns*

Izu, C., Mirolo, C., Settle, A., Mannila, L., Stupuriene, G. (2017) "Exploring Bebras Tasks Content and Performance: A Multinational Study.," *Informatics in Education*, 16(1), 39–59.

Jenkins, C. (2015) "Poem Generator: A Comparative Quantitative Evaluation of a Microworlds-Based Learning Approach for Teaching English," *International Journal of Education and Development using Information and Communication Technology*, 11(2), 153–167.

Kitchenham, B., Charters, S. (2007) "Guidelines for performing systematic literature reviews in software engineering," Engineering, 2 (EBSE 2007-001).

Koh, K.H., Basawapatna, A., Nickerson, H., Repenning, A. (2014) "Real time assessment of computational thinking," in *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, Melbourne, VIC, Australia, 49 – 52.

Korkmaz, Ö., Çakir, R. and Özden, M.Y., 2017. A validity and reliability study of the Computational Thinking Scales (CTS). Computers in Human Behavior, 72, pp.558-569.

Krugel, J., Hubwieser, P. (2017) “Computational thinking as springboard for learning object-oriented programming in an interactive MOOC,” in *IEEE Global Engineering Education Conference, EDUCON*, Athens, Greece, 1709 – 1712.

Liao, L., Liang, J. (2017) “An empirical study on blended learning to promote the development of computational thinking ability of college students,” in *2017 International Symposium on Educational Technology (ISET). Proceedings*, Los Alamitos, CA, USA, 256 – 60.

Marcelino, M.J., Pessoa, T., Vieira, C., Salvador, T., Mendes, A.J. (2017) “Learning Computational Thinking and scratch at distance,” *Computers in Human Behavior*.

Marshall, K.S. (2011) “Was that CT? Assessing Computational Thinking Patterns through Video-Based Prompts,” *In Proceedings of the 2011 Annual Meeting of the American Educational Research Association (AERA) (New Orleans, LA, April 8-12, 2011)*.

Menekse, M. (2015). Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. Computer Science Education, 25(4), 325-350.

Miller, L.D., Soh, L.-K., Chiriacescu, V., Ingraham, E., Shell, D.F., Ramsay, S., Hazley, M.P. (2013) “Improving learning of computational thinking using creative thinking exercises in CS-1 computer science courses,” in *Proceedings - Frontiers in Education Conference, FIE*, Oklahoma City, OK, United states, 1426 – 1432,

Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G. and Prisma Group, 2009. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. PLoS medicine, 6(7), p.e1000097.

Mouza, C., Yang, H., Pan, Y.-C., Ozden, S.Y., Pollock, L. (2017) “Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK).,” *Australasian Journal of Educational Technology*, 33(3), 61–76.

National Research Council (NRC) (2010) *Report of a Workshop on the Scope and Nature of Computational Thinking*, National Academies Press.

Othman, M., Zain, N.M., Mazlan, U.H., Zainordin, R. (2015) “Assessing cognitive enhancements in introductory programming through online collaborative learning system,” in *2015 International Symposium on Mathematical Sciences and Computing Research, iSMSC 2015 - Proceedings*, Bandar Meru Raya, Ipoh, Malaysia, 7 – 12, Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, Inc. Pellas, N., Peroutseas, E. (2016) “Gaming in Second Life via Scratch4SL.,” *Journal of Educational Computing Research*, 54(1), 108–143,

Repenning, A., Basawapatna, A., Koh, K.H. (2009) “Making University Education More Like Middle School Computer Club: Facilitating the Flow of Inspiration,” in *Proceedings of the 14th Western Canadian Conference on Computing Education*, WCCCE ’09, ACM: New York, NY, USA, 9–16.

Sener, J. 2015. Definitions of E-Learning Courses and Programs Version 2.0 April 4, 2015 Developed for Discussion within the Online Learning Community By Frank Mayadas, Gary Miller, and John Sener. Online Learning Consortium. Available from: https://onlinelearningconsortium.org/updated-e-learning-definitions-2/.

Shell, D.F., Hazley, M.P., Soh, L.-K., Dee Miller, L., Chiriacescu, V., Ingraham, E. (2014) “Improving learning of computational thinking using computational creativity exercises in a college CSI computer science course for engineers,” in *2014 IEEE Frontiers in Education Conference (FIE). Proceedings*, Piscataway, NJ, USA, 1 – 7. Shute, V.J., Sun, C., Asbell-Clarke, J. (2017) “Demystifying computational thinking,” *Educational Research Review*, 22, 142–158.

Tsai, C.-W., Shen, P.-D., Tsai, M.-C., Chen, W.-Y. (2017) “Exploring the effects of web-mediated computational thinking on developing students’ computing skills in a ubiquitous learning environment.,” *Interactive Learning Environments*, 25(6), 762–777.

Tsai, M.-C., Tsai, C.-W. (2017) “Applying online externally-facilitated regulated learning and computational thinking to improve students learning,” *Universal Access in the Information Society*, 1 – 10. Vivian, R., Falkner, K., Falkner, N. (2014) “Addressing the challenges of a new digital technologies curriculum:
MOOCs as a scalable solution for teacher professional development.,” *Research in Learning Technology*, 22, 1–19. h Weintrop, D., Holbert, N., Horn, M.S., Wilensky, U. (2016) “Computational Thinking in Constructionist Video Games,” *Int. J. Game-Based Learn.*, 6(1), 1–17.

Wilkerson-Jerde, M. (2014) “Construction, categorization, and consensus: student generated computational artifacts as a context for disciplinary reflection.,” *Educational Technology Research \& Development*, 62(1), 99–121.

Wing, J.M. (2006) “Computational Thinking,” *Communications of the ACM*, 49(3), 33–35. Wing, J.M. (2008) “Computational thinking and thinking about computing.,” *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 366(1881), 3717–25.

Wing, J.M. (2011) Research Notebook: Computational thinking—What and Why? The Link Magazine, Spring [online], available: https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why. Xie, B., Abelson, H. (2016) “Skill progression in MIT app inventor,” in *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, Cambridge, United kingdom, 213 – 217.

Yeh, K.C., Xie, Y. and Ke, F., 2011, October. Teaching computational thinking to non-computing majors using spreadsheet functions. In Frontiers in Education Conference (FIE), 2011 (pp. F3J-1). IEEE.