A Literature Review: Fostering Computational Thinking Through Game-Based Learning in K-12

Cansu Tatar  
North Carolina State University  
Department of Teacher Education and Learning Sciences  
402F Poe Hall  
Raleigh, NC 27695  
Phone: 984-292-966  
Email: ctatar@ncsu.edu

Deniz Eseryel  
North Carolina State University  
Department of Teacher Education and Learning Sciences  
402 Poe Hall  
Raleigh, NC 27695  
Phone: 919-513-8587  
Email: deserye@ncsu.edu

Introduction

Computational thinking (CT) is defined as an analytical thinking approach and problem-solving process that breaking problems into small pieces, designing systems for possible solutions, and understanding the relationship among the pieces by drawing fundamental concepts (Wing, 2006, 2008; National Research Council, 2010a). When Wing’s (2006, p. 33-35) pronouncement that “[computational thinking] represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use” (p. 33), following this trend, the number of studies related to computational thinking and its’ implementations shows a huge increase. However, researchers are usually confused with the difference between CT and computer science (CS) concepts. Even though CT consists of some concepts that are the core of computing and computer science, it also includes efficient practices such as abstraction, decomposition, pattern generalization, and prediction. These concepts are also used by scientific and mathematical disciplines for modeling, reasoning, and problem-solving (National Research Council, 2008). According to Henderson et al. (2007), CT has been shown as the core of all science, technology, engineering, and mathematics (STEM) disciplines and it leads the idea of teaching computing in K-12 education. Also, most of the studies show teaching programming and CT skills is an effective method to cope with the challenging science and math concepts (Hambrusch et al., 2009, p. 183-187; Blikstein and Wilensky, 2009, p. 81-119).

After Wing’s call to action for teaching CT skills to prepare students for being a productive member of the society, two National Academy of Sciences workshops were organized to find out the nature of computational thinking and its cognitive and educational interventions on K-12 curriculum (National Research Council, 2010b). These workshops also aimed to explain the pedagogical aspects of CT (National Research Council, 2011). In the first CT workshop, procedural thinking and programming (Papert, 1980, 1981, p. 82-92) were discussed as early concepts of CT. Even though this workshop seemed like a beginning process, it added significant value to examine and evaluate many concepts from computer science discipline to take CT beyond the idea of “just programming”. On the other hand, the workshop found out the lack of consensus that looks like to have bedeviled this field. Besides, it led to many unanswered questions: “How can CT be recognized? What is the best pedagogy for promoting CT among children? Can programming, computers, and CT be legitimately separated?” (National Research Council, 2010)

While earlier researchers focus on the definitions of CT, since 2012, research interests related to CT have been shifted to work on these unanswered questions. Graver and Pea (2013, p. 38-43) explain that researchers have started to interest more practical questions such as how to encourage the implementation of CT in K-12 education and how to assess students’ CT abilities. In the last three decades, issues of how to teach and learn programming and CS constitute a large amount of the literature. Much of what we collectively know is largely based on Papert’s studies in the 1980s by using game-based learning environments using languages such as LOGO and BASIC. He indicates that “[we] can give children unprecedented power to invent and carry on exciting projects by providing them with access
to computers, with a suitably clear and intelligent programming language and with peripheral devices capable of producing on-line-real-time action” (Papert, 1972, p. 245-255). Yet, providing all sources without knowing how to implement them in K12 settings to increase students’ motivation and learning achievement should be answered. However, the bulk of this CS education research is set in the context of undergraduate classrooms; little is known about how applications of game-based learning foster computational thinking in K-12, especially in elementary schools.

Therefore, the purpose of this study is to obtain a better understanding for educators and researchers by systematically reviewing and synthesizing recent research since 2010 on CT regarding applications of game-based learning (GBL) in K-12.

The following research questions are addressed in this study.

RQ1: What range of topics does the current literature cover?
RQ2: What kind of game-based learning environments and tools have been shown effective in promoting CT?
RQ3: What are the reported outcomes related to students’ engagement, motivation, and achievement in CT?
RQ4: How was the CT skills assessed in recent literature?

Methodology

Data Sources

The main data sources include journal articles, conference papers, and doctoral dissertations. However, due to language barriers, only articles written in English were included. Also, the discipline criteria are limited to the field of education because of researchers’ academic background. The following keywords and their combination were used with regard to the main concepts of CT and GBL by using Boolean logic such as (Computational thinking AND (k12-K12 OR elementary school OR middle school)).

The search was conducted in databases that are well-known and well-established in the field of computational thinking and game-based learning: ERIC (Education Resources Information Center), PsycInfo, IJGBL (International Journal of Game-Based Learning), IEEE (Institute of Electrical and Electronics Engineers), and Summon. Besides, Google Scholar was used for additional searches.

Keywords:
• Computational thinking
• Computing education
• Game-based learning
• Games
• Gamification
• Digital games
• k12 or K12 or Elementary school or Secondary Education
• STEM education
• Non-STEM education

Coding and Analysis

The first step of the study identified 951 articles between 2010 and 2019 (see Table 1). At the beginning of the coding and analysis process, these studies were examined initially in regard to their titles, abstracts, and keywords. The initial review helped to see the relatedness of the studies at first. The criteria of relatedness were whether studies provided analytical results related to enhance computational thinking or not (see Table 2).

Table 1. Distributions of studies in each database

<table>
<thead>
<tr>
<th>Database</th>
<th>Number of Articles Identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERIC</td>
<td>13</td>
</tr>
<tr>
<td>PsycInfo</td>
<td>3</td>
</tr>
<tr>
<td>IJGBL</td>
<td>649</td>
</tr>
</tbody>
</table>

289
In the selection stage, different frameworks were followed used by Zhang and Nouri (2019) and Sengupta et al. (2012).

<table>
<thead>
<tr>
<th>Table 2. Selection criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inclusion Criteria</strong></td>
</tr>
<tr>
<td>K12 education (kindergarten through the 12th grade (1-12))</td>
</tr>
<tr>
<td>Empirical studies</td>
</tr>
<tr>
<td>Studies that provide evidence about students’ learning outcomes</td>
</tr>
<tr>
<td>Using educational tools or technologies to foster computational thinking development</td>
</tr>
</tbody>
</table>

**Findings**

**Computational Thinking Through Game Based Learning**

Game-based learning is defined as implementing game elements such as core mechanics, challenges, and goals into real-life settings to enhance learning. According to the study of the National Purchase Diary Panel (NPD) Group, in the United States, %82 of children ages between 2 to 17, approximately 64 million children, play video games (N.P.D Group, 2009). This study reveals how much games are attractive to children. Therefore, GBL as a teaching method seems promising to enhance students’ learning and motivation. Since an increase in implementation of CT pushes schools searching effective teaching methods (Gouws, Bradshaw, and Wentworth, 2013, p. 10-15), many researchers indicate that GBL, especially game design part, seems promising for teaching CT concepts in K-12 settings (Baytak and Land, 2010; Rowe et al., 2017, p. 45).

In the literature, different theoretical frameworks are promoted to guide students and teachers to improve CT skills through GBL. However, these frameworks show differences based on teaching or focusing particular skill(s): decomposition, pattern recognition, abstraction, algorithm thinking, and evaluation (for checking the CT skills see Selby and Woollard, 2013) in the development of CT. In Figure 1, Jiang et al., (2019, p. 29) suggest a theoretical framework for targeting coding environments to improve CT skills with regard to gameplay elements and design principles. In the framework, they put emphasis on computational problem-solving (CPS) that a basis aspect of CT, design of a puzzle game, and their relationship.

![Figure 1. Computational Puzzle Design (CPD) Framework (Source: Jiang et al.,2019, p. 29)]
Many studies especially focus on block-based coding game designs, follow similar structured frameworks like CPD, even though they do not specifically mention their theoretical framework. In these studies, players’ tasks and game designs show differences but the relationship between CPS and game elements is similar (e.g. Troiano et al., 2019, p. 208-219; Kazimoglu et al., 2012, p. 522-531; Harrison et al., 2018, p. 134-138).

Following a structured theoretical framework is not solely enough for the development of CT skills, game elements and design in regard to the target audience play a significant role in students’ learning outcomes. In the literature, many researchers develop their own game projects while others prefer to use already developed software such as LEGO Mindstorm, Scratch, and Light Bot. However, surprisingly, even though researchers develop their own game designs, still, they tend to use block-based game design methodology (e.g. Howland and Good, 2015, p. 224-240; Leonard et al., 2016; Bauer, Butler, and Popović, 2017, p. 26; Jiang et al., 2019, p. 29). Bauer et al., (2017, p. 26) explain their reason to develop their own game design as keeping a balance between exploration and guidance in the games. According to them, existing game designs (called Dragon Architect) that aim to improve CT include either open-ended exploration with little direct instruction or full of exploration with a lot of direct instruction. In their game design, players use code blocks to move the dragon character for solving the puzzle (see Figure 2). Similarly, Jiang et al., (2019, p. 29) promote the “LittleWorld” game that includes non-coding blocks and puzzle designs to foster students’ computational problem-solving skills (see Figure 2). They indicate that puzzle game designs are appropriate for enhancing logical thinking, problem-solving skills, and pattern recognition which CT aims to promote.

![Figure 2. Examples of block-based game designs (Dragon Architect and LittleWorld) (Source: Bauer, Butler, and Popovic, 2017)](image)

Moreover, a few studies implement hands-on game-based methods and role-playing game designs to develop CT skills and to enhance student communication, reading, and writing skills. For example, Jagušt et al. (2018, p. 1-5) developed unplugged game-based activities to teach CT skills. In their study, four case studies were conducted with different age groups. However, only three of them are related to K-12 students, so the details of the fourth case study wouldn’t be provided here. In the first case study, they redesigned the famous children games such as Battleship, the dot game, and Packman by using pencil and paper. In this CT activity, students completed basic paper programming activities like doing in Scratch. Similarly, in the second case study, students completed a life-size graph paper game by working either individualist or collaboratively. In the single player mode, a student should execute the given program (drawing a shape) in the playing board. In the two-player mode, one student takes a “programmer” role while other plays as “robot”. The programmer writes the codes to complete the given programming task while the robot executes these written commands. Furthermore, the third case study is called “The Network”. The purpose of this study is to integrate the networks problems and unplugged activities into the curriculum. 6th grade students completed these activities like in the first case study.
In addition to Jagušt et al. (2018, p. 1-5), another GBL activity is conducted to explore the relationship between computational thinking and English literature in the K-12 curriculum by Nesiba, Pontelli, and Staley in 2015. In their study, existed English curriculum units are integrated with GBL methods and CT skills. For example, students practice their CT skills by telling the story of “Macbeth” novel, the story of a Scottish soldier who wants to be King without thinking any consequences during his journey (see Figure 4). At first, students select some scenes from the novel and sketch out them for preparing a storyboard. After they create their storyboard, they implement these scenes by using ToonDoo tool, online comic-creation tool. Even though this study does not include any problem-solving skills, students have to use their algorithmic thinking abilities to develop a detailed step by step scene based on their storyboard. In the literature, algorithmic thinking is described as a way of getting to the solution by defining clear steps (Selby and Woollard, 2013; Edwards, 2011, p. 58-67). Although students do not solve any problem in the study, they define each step clearly to complete their tasks. Thus, this study provides an exceptional example of how to foster algorithmic thinking without a problem-solving task. Besides, according to Nesiba, Pontelli, and Staley (2015, p. 1-8), students also use their abstraction skills while scanning their scenes and criticizing the significant passages. However, the abstraction component of CT skills is defined as identifying general principles to generate specific patterns (Shute, Sun, and Asbell-Clarke, 2017, p. 142-158). Therefore, their findings related to abstraction and decomposition is arguable since students do not focus on generating patterns in their study.

Overall, most of the studies in terms of implementation of CT through GBL in K-12 setting address coding-oriented education or STEM education by using block-based applications such as Scratch, LightBot, and Combats. Only a few studies aim to enhance students’ CT skills in social science classes.

**Students’ Engagement, Motivation, and Achievement in CT**

According to Kotini and Tzelepi (2015, p. 219-252), the way of enhancing students’ CT skills is complex and is dependent on students’ engagement and motivation. Also, they indicate that gamified learning activities aim to foster CT skills have a positive impact on students’ achievement. Following this trend, many studies show integration of educational games into the K-12 curriculum help to increase students’ attention, learning motivation, and positive attitude (Yang and Wu, 2012, p. 339-352; Bai et al., 2012, p. 993-1003; Triantafyllakos, Palageorgiou, and Tsoukalas, 2011, p. 227-242). However, only a few studies mention about improvements in students’ motivation in the implementation of a game-based learning activity to foster students’ CT skills. One example from these studies is conducted by Fogli et al. in 2017. They studied with 18 secondary school students to measure the effectiveness of a game-based system (TAPASPlay), which is designed as a block-based environment to enhance students’ CT skills.
According to their study, there is evidence that TAPASPlay provides an enjoyable environment to improve students’ motivation and CT skills with regard to collaborative learning. Moreover, a study also shows that primary school students are highly motivated to learn algorithmic thinking with tangible objects by using block-based environments like BYOB or Scratch (Futschek and Moschitz, 2011, p. 155-164). These two studies advocate that learning CT skills through GBL, or tangible objects can increase students’ motivation and engagement. On the other hand, this kind of studies is insufficient to explain the source of students’ motivation by showing evidence. In other words, studies in the literature do not provide detailed information about either student get motivated by involving a game-based activity or they are motivated because of improving their CT skills.

Furthermore, in terms of students’ achievements in CT, most of the studies only put emphasis on decomposition, algorithm design, and abstract thinking (Harrison et al., 2018, p. 134-138; Kazimoglu et al., 2012; Bauer et al., 2017, p. 26). Surprisingly, current studies do not usually mention about pattern recognition and generalization while talking about students’ CT skills. However, these two CT components are the basis for the development of other components like algorithm design (Ambrosio et al., 2014, p. 25-34). In general, more studies should be conducted to explain the effects of CT through GBL on students’ engagement, motivation, and achievement in particular CT components.

**Assessment and Evaluation Methods in CT**

Adaptation of CT into the school curriculum is essential to prepare students for 21st-century skills and to be productive members of the society. However, it requires an assessment framework to understand in what extent K12 school students are successful in CT abilities and which kind of GBL methods can engage students’ CT skills. Also, Nitko and Brookhart, (1996) indicate that assessment aims to improve the learning process with regard to needs of the students, schools, and curriculum. Besides, assessment tools play a critical role to bring CT through GBL to K-12 settings for supporting the use of programming in middle school (Werner et al., 2012, p. 215-220) (Grover and Pea, 2013, p. 38-43; Grover, 2015, p. 15-20). However, a widely accepted assessment method is missing to measure the effectiveness of CT interventions in the literature (Shute, Sun, and Asbell-Clarke, 2017, p. 142-158). According to Settle et al., (2012, p. 22-27), the needs for an appropriate assessment method is a critical step to control the reliability and validity of interventions in the CT area.

In the literature, most of the studies use interviews, pre and post-tests, project portfolio analysis, document-based analysis, design scenarios or the combination of these methods as assessment methods in the interventions of CT (Brennan and Resnick, 2012, p. 25; Bubica and Boljat, 2018; Mioto et al., 2019). These are the traditional ways to gain an understanding of students’ learning outcomes related to CT concepts. Additionally, Shute et al., (2017, p. 142-158) present two assessment methods: Scratch-based and game/simulation-based. According to their study, Scratch-based assessment also includes portfolio evaluation, interviews, and design projects in addition to multiple choice and open-ended quizzes. On the other hand, they bring up pattern analysis and web-like graphic reports in the game/simulation assessment method. This evaluation method is based on visual analysis of students’ progress.

Furthermore, Román-González, Moreno-León and Robles (2019) propose a comprehensive evaluation model to assess students’ CT abilities in regards to Bloom’s taxonomy (see Figure 5). They suggest several tools such as diagnostic, summative, and formative-iterative. Each tool specifically addresses one level of CT. For example, diagnostic tools aim to measure the understanding level of CT while summative tools focus on learning objectives. According to Mioto et al., (2019), since formative and summative assessments put emphasis on students’ cognitive abilities, they are more efficient during the development of CT. However, a few studies use these evaluation tools and do not include concrete examples of how to combine these methods in educational settings (Román-González, Moreno-León and Robles, 2019).

To sum up, both qualitative and quantitative methods are preferable for the researchers in the field. Only, a few studies get benefits from data visualization reports to measure students’ growth. The examples of how to implement different assessment methods into CT education can take place in the future studies.
Limitations of the Study

This study has two main limitations. Firstly, the review is only conducted searches in five databases and Google Scholar. Also, studies are only written in English and the field of education was eligible. Therefore, it might not represent all works in the research area.

In addition, this review focuses on the studies which provide positive evidence to show how students’ CT abilities can be enhanced through GBL. Thus, this scope may not present a complete picture of the improvement of CT skills in K12 education. It might lead to publication bias as a limitation of this study.

Conclusion and Discussion

Space limitations of this proposal limit our ability to discuss our synthesis in detail, which we intend to present at the AECT conference organized by the four research questions followed by recommendations for future research. Below are some of the highlights for each research question.

According to our findings, most studies have focused on either STEM education to children in regard to computational thinking. In other words, most of the studies consider STEM components to increase learning outcomes, students’ confidence, and improve their capacities in terms of enhancing computational thinking. However, studies mainly put emphasis on the science component even though they come up with a generalization in STEM. For example, Jenson and Droumeva, (2016, p. 111-121) start with STEM as the main concern but then, they make narrow the study by putting a spotlight on science concept. On the other, a few studies only consider all of the STEM subjects in their research methodology. For instance, Tsarava et al., (2017, p. 687-695) focus on science, technology, engineering, and math subjects by building a bridge between real-life problems and CT-based solving. They design eight lessons that consider different problems related to each STEM subject for 3rd and 4th graders in primary schools. Their study is one of the few studies that show how to foster computational thinking through game-based learning in different subjects. Nevertheless, more study, especially related to technology and engineering in STEM, should be conducted to guide teachers or policy makers on how to enhance CT through GBL in these subjects. Besides, the number of studies in non-STEM disciplines like literacy or social sciences is also a few. Researchers can consider working on how to develop a CT curriculum to enhance students’ communication, reading, and writing skills.

In addition, from a game-based learning perspective, our findings show that most of the studies regard block-based environments to support CT. Most common trends in the studies in terms of block-based environments are Scratch Jr, MIT AppInventor, Game Maker, Code Combats, ColoBot, and LEGO Programs. These environments are usually used for teaching algorithm or creating lesson plans to blend in an ongoing lesson for enhancing CT. For example, Baratè, Ludovico, and Malchiodi (2017) used the LEGO-based music notation program to foster primary
school students’ CT abilities. The findings of these studies emphasize a strong relationship between basic object-oriented programming and computational thinking. Moreover, from a theoretical perspective, we found that several studies are based on constructionism, constructivism, project-based theory, and active learning theory. Future studies might want to consider the role of social cognitive theory, information processing theory, and cognitive learning processes on students’ CT abilities. Since CT includes relevant abilities such as decomposition, the process of debugging, testing and analysis, and control structures are related to these learning theories.

Most recent addressing the CT assessment has used either student-created, or pre-designed programming artifacts to evaluate students’ understanding and use of abstraction, conditional logic, algorithmic thinking, and other CT concepts to solve problems. In addition, ideas of deconstruction, reverse engineering, and debugging was mostly used to assess children's understanding in computational contexts.

The findings of this study also pointed out to the use of gamification as a way to foster students’ computational thinking abilities, engagement and motivation in addition to their academic achievement at early ages.

The analysis of existing literature in this study demonstrates that using game-based learning as an interdisciplinary activity has the potential to foster computational thinking and develop 21st-century competencies such as critical thinking, problem-solving, creation, and innovation. However, current studies did not focus on the roles of complexity, metacognition, and fluid intelligence for facilitating CT abilities through GBL. Thus, we identified those areas for further studies.

References


Fogli, D., Danesi, F., Malizia, A., Turchi, T., & Bell, D. (2017). Sustaining cultures of participation by fostering computational thinking skills through game-play. Paper presented at the GHITALY@ CH, Italy,


