Exploring the Development of Primary School Students' Computational Thinking and 21st Century Skills through Scaffolding: Voices from the Stakeholders

Volkan Kukul[1]

Recep Çakır[1]

[1]Amasya University

Abstract

Computational thinking (CT) has become a skill that is taught starting from an early age with its increasing popularity. In addition, the opinion that CT is related to other 21st century skills finds its place in the literature. The main purpose of this study is to identify the contributions of scaffolding-based game programming activities to students' CT and 21st-century skills. In line with this purpose, the study was designed as a qualitative case study. The participants of the study consisted of 16 primary school students, 10 pre-service ICT Teachers, and 2 primary school teachers. The research results reveal that game programming has a positive effect on students' CT skills, 21st century skills and some psychometric variables like self-confidence and motivation. In future studies and implementations, educators may support their instructions of programming through different scaffolding strategies. In addition, it should be taken into consideration that students can become innovative designers with content that they find interesting.

Keywords: Computational Thinking, 21st Century Skills, Scaffolding, Game Programming

1. Introduction

In modern societies, people need diverse skills to maintain their lives more conveniently. These skills can vary according to the approaches of institutions and researchers, yet there are common points between the skills that are defined. The common points involve creative, innovative, collaborative and computational thinking skills (PFC, 2018; van Laar, van Deursen, van Dijk, & de Haan, 2017; Chalkiadaki, 2018). Teaching these skills requires updating education programs along with the advancing technology and enabling students to perform the necessary activities. Coding skills and programming education have taken place among the rapidly-growing trends with the advancing technology.

Coding skills are known as the skills that emerge as an outcome of reasoning and they are given a place in the study programs of different countries (Angeli et al., 2016; European Commission, 2016). In addition to computer programming skills, students are also taught skills to find different solutions to problems and find out practical answers (Yukselturk and Altiok, 2016). The aim is to raise number of students who can use technology well and develop their own technology.

Recent studies have shown that children can learn coding even at early ages and coding education improves the skills that students should have in the 21st century (Bermingham et al., 2013; Zimmerman, 2007). Cetin (2012) has shown in his study that was carried out with 5th-grade students that teaching coding to students had a positive impact on their problem-solving skills. Another study conducted by Lin and Kuo (2010) employed the

robot kits for education in teaching the programming language course at primary and secondary school levels, and their study reported that students can gain programming skills with these kits and acquire basic architectural knowledge. Lindh and Holgersson (2007) investigated the impacts of robotic toys (legos) on mathematics and problem-solving skills in their study, yet did not find a significant difference between the control and experimental groups. However, the study revealed that students who enjoyed problem-solving activities had higher levels of achievement, and students who had lego-logo training became more successful in the following years.

In the initial phases of the implementation, students who are new to programming education had considered programming education as a difficult process (Genc and Karakus, 2011; Gomes and Mendes, 2007), yet, recently, programming education has become more entertaining for students with ready codes and visualized tools (code hour, code org, scratch, etc.) and perceived as less difficult by the learners. One of the approaches adopted to make programming education an entertaining process for students is enabling them to design their own games. Programming games is reported as an effective method to teach technical skills to students (Kafai and Burke, 2015; Hainey, Baxter, & Ford, 2019). Kafai and Burke (2015) argued that coding education can reach to a larger audience and become more successful with games and suggested to teach coding via games. Furthermore, in the literature, game programming is defined as one of the most common approaches to improve computational thinking skill (Weinberg, 2013; Hainey et al., 2019; Allsop, 2019) which is among the 21st-century skills (ISTE, 2016). Therefore, this aim of this study is to analyze the impacts of game programming on 21st-century skills and computational thinking skills from the perspectives of all relevant stakeholders.

2. Theoretical Framework

2.1 21st Century Skills

Researchers have emphasized that individuals should be equipped with different skills to be able to accommodate themselves in knowledge societies. Digital literacy lay at the core of these skills (Perkovic and Settle, 2010). Considering the globalization process and the rapid development of ICTs in our lives, improving digital skills has become an imperative (Van Laar, Van Deursen, Van Dijk, & De Haan, 2017). According to Perkovic and Settle (2010), to catch up with 21st-century societies, individuals should be computer literate and possess computer fluency ve computational thinking skills. On the other hand, for various organizations, 21st-century skills are not only based on digital literacy and they should be viewed from a broader perspective. Partnership for 21st Century Skills analyzed the skills that individuals must have in this century under three main themes in the framework published in 2007 (Partnership for 21st Century Learning, 2007). In framework, the first theme is Learning and Innovation Skills (creativity and innovation, critical thinking and problem solving, communication and collaboration), the second theme is Information, Media and Technology Skills (information, media and ICT literacy), and the third theme is Life and Career Skills'tir (flexibility and adaptability, initiative and self-direction, social and cross-cultural skills, productivity and accountability, leadership and responsibility). International Society for Technology in Education (ISTE) has also established 21st century standards for students in different periods. First of all, in 1998, they improved individuals' use of technology based on their learning, afterward they developed standards for using technology for learning, and finally, they developed standards for transformative learning with technology in 2016. The recent standards include the skills as follows: Empowered Learner, Digital Citizen, Knowledge Constructor, Innovative Designer, Computational Thinker, Creative Communicator and Global Collaborator (ISTE, 2016).

2.2 Computational Thinking

The concept of computational thinking (CT) was used by Papert for the first time in the 1980s and it was brought up again by Jeannette Wing in 2006 (Kalelioglu, Gulbahar, & Kukul, 2016; Kukul & Karatas, 2019; Roman-Gonzalez, Perez-Gonzalez, & Jimenez-Fernandez, 2017). After that date, many countries have updated their curriculums to improve students' computational thinking skills (Mannila et al., 2014). For a long time, there was no consensus on the definition of CT (Grover & Pea, 2013; Kalelioglu et al., 2016). The definition of Wing, which made the concept of computational thinking popular again which defines the concept as "involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science" (Jeannette M. Wing, 2006). However, the article which presented this definition also raised many questions. In 2010, Wing has clarified the definition of CT in another study and reformulate it as CT is the "thought processes involved in formulating problems and their solutions so that the solutions are represented

in a form that can be effectively carried out by an information-processing agent" (J. Wing, 2010, p.1). The given definitions were not found sufficient to present the structure of CT skill as a psychological variable, and therefore to be evaluated and improved (Roman-Gonzalez, Perez-Gonzalez, Moreno-Leon, & Robles, 2018). Therefore, this need has tried to be fulfilled with studies on definitions towards implementation (Kalelioglu et al., 2016) and evaluation (Brennan & Resnick, 2012; Dolgopolovas, Jevsikova, Savulionienė, & Dagienė, 2015; Kukul & Karatas, 2019; Román-González, 2015; Allsop, 2019; Hainey et al., 2019). According to Weinberg (2013), the updates in the curriculum based on teaching CT skills and the most common approach can be addressed under four main headlines. These approaches include block-based programming environments (Scratch, AppInventor, Alice, etc.), game programming (KoduGameLab, etc.) or Robotic (Arduino, Lego, etc.) applications, computer-less applications (CS Unplugged, Bebras, etc.) and CT with other disciplines (STEM, etc.). In the scope of this study, a game programming approach, which is a widely used approach in teaching the CT skill, was used.

2.3 Game Programming and Kodu Game Lab

In the literature, some studies focused on the potential impacts of game programming on CT (Lee et al., 2011), and it is evident that game programming improves CT skills (Garneli & Chorianopoulos, 2018). The meta-synthesis study conducted by Denner, Campe, and Werner (2019) revealed that students improved diverse skills while programming games and also enjoyed the process of programming. Vos, van der Meijden, and Denessen, (2011) compared two groups of students (aged 10-12 years old). One group only played games and the other group created their own games. According to the results, the learning motivation of the students in the group who created their own games is higher than the others. The given situation makes game programming a strong alternative for enhancing CT skills. Technology firms are also aware of this fact and have created platforms where children can develop games. KoduGameLab, which was used in the scope of this study, is an example of these platforms.

KoduGameLab is an environment that was developed by Microsoft that enables to program games based on the drag-and-drop system. The system has been designed due to the increasing popularity of teaching programming to young children, as an environment where children can design their own games. The program was introduced as a desktop version and it can run with different operating systems. The tool that has object-oriented programming features (Yukselturk & Uçgul, 2018) also includes libraries consisted of different game environments, and individuals can choose one of the environments and use different components. It is possible to share the game designed with KoduGameLab with others, and also to access other games that are shared. KoduGameLab enables the design of three-dimensional games and in this environment, the games are designed using condition structures frequently. In the literature, several studies are designed by using KoduGameLab for game programming. The experimental study conducted by Akcaoglu and Green (2019) revealed that games developed with KoduGameLab improve students design skills. Fowler and Khosmood (2018) conducted a study that employed KoduGameLab and showed that game programming had a positive impact on students' perceptions of Computer Science and permanent learning abilities. Chiazzese et al. (2018) revealed in their study that game programming had a positive impact on participants understanding of the basic principles of programming.

2.4 Scaffolding

The concept of Scaffolding was defined by Vygotsky for the first time in 1978 (Dennen, 2004). According to Vygotsky, students should be guided by a more talented peer to overcome a problem or fulfil a task (Vygotsky, 1978). From this aspect, the concept can be considered as a strategy that fits to group studies. Scaffolding strategies, which are based on learners' needs, may increase achievement in learning concepts, procedures and metacognitive skills (McLoughlin, 2002). In addition to metacognitive skills, the concept also plays a role in increasing achievement in affective development spheres such as motivation and self-confidence (Dennen, 2004). It can also support students to overcome their feelings of failure (Bean & Patel Stevens, 2002). In principle, various strategies such as modelling, coaching, prompting and questioning are used for scaffolding implementations based on social dialog and interaction (Ge & Land, 2003). It is believed that the concept of scaffolding implemented with these strategies may contribute to CT components. For example, Rogoff (1990) defined scaffolding as a meta-activity that assists learners to separate given tasks into pieces, and this definition can explain supporting learners in 'decomposition' which is among the sub-components of CT.

Considering the age levels and lack of programming experience of the students who participated in this study, implementation of scaffolding strategies with talented and experienced peers on programming, and with pre-service teachers will be useful for students. Feng and Chen (2014) emphasized that scaffolding strategies that will be implemented with learners can enable students to analyse the problem effectively before solving them. From this aspect, scaffolding may also contribute to the abstraction skill that is seen as the most important component of CT (Jeannette M Wing, 2008).

3. Aim of the Study

CT has become a skill that is taught starting from an early age with its increasing popularity. In the literature, scaffolding-based programming education has been emphasized as a frequently used method (Lye & Koh, 2014), in the evaluation of these studies computational concepts such as variables and loops were employed (e.g. Kazakoff & Bers, 2012; Wang & Chen, 2010), yet, computational practices (problem-solving practices during programming activities (Brennan & Resnick, 2012) was not stressed (Lye & Koh, 2014). The research studies conducted on the issue suggested evaluating students in applications/practices that aim to teach the CT skill to students through think-aloud protocol or observation of the student outcomes (Lye & Koh, 2014). The main purpose of this study is to identify the contributions of scaffolding-based game programming activities to students' CT and 21st-century skills. In line with this purpose, answers were sought to the following research questions:

- Considering the CT components, which components have been affected by the contributions of game programming activities?

- What were the contributions of game programming activities to students' 21st-century skills? In the case that there were contributions, which skills are affected by these contributions and in which way they were affected?

4. Method

The purpose of the research study is to identify how game programming may affect students' learning of CT and 21st-century skills. In line with this purpose, the study was designed as a qualitative case study. Creswell (2007) defined a case study as "a qualitative approach in which the investigator explores a bounded system (a case) or multiple bounded systems (cases) over time, through detailed, in-depth data collection involving multiple sources of information". The case that was chosen in the scope of this study was the game programming activities organized by STEM and Coding Student Club and performed by students via KoduGame Lab as a part of the end of year activities of a primary school located in Amasya province. In the scope of this activity, students programmed their games in the framework of the selected theme under the guidance of Information Technologies teachers. The data sources of the study included the interviews conducted with the students who participated in the activity, with the teachers who were in charge of the student clubs, and with pre-service Information Technologies teachers who guided students. The observations made by pre-service teachers were also used in this process. The 'computational thinking evaluation framework' developed by Brennan and Resnick (2012) constituted the theoretical basis for the interview questions, given that the study aimed to evaluate students' CT and 21st-century skills via programming.

4.1 Participants

The participants of the study consisted of 16 primary school students of a public school located in Amasya province who participated on the activities organized by STEM and Coding Student Club, two teachers who were in charge of the students club, and 10 pre-service Information Technologies teachers who were in the third year of study. The distribution of the students according to the grade-levels showed that 4 students were second-grade students, 6 students were third-grade students, and six students were fourth-grade students. Some of these students previously joined coding activities organized by the club in the previous year and therefore they were experienced in coding, however, the majority of the students participated in club activities for the first time. On the other hand, the teachers who were in charge of the clubs were class teachers and had teaching experience over 20 years. The pre-service Information Technologies teachers who guided students participated in this study voluntarily on the scope of the practical social work course.

4.2 Content

Due to the inadequate physical capacities of their school, students used the computer laboratories of Amasya University for the club activities. In the first phase, students were separated into groups by the teachers who were in charge of the clubs, and afterward, at least one teacher was assigned to each group to guide students. Before introducing the university environment to the students, basic knowledge on the KoduGameLab interface and the logic of the system was provided to the students via online education programs. In the university environment, the students were asked to design a game related to the 'space' theme that was defined by the teachers. Students coded their games for 9 weeks and two hours were allocated per week for the coding activities. As a result of the activities, the students presented their games within Amasya University. Some of the games that were created by students were selected to take part in a national competition. The pre-service teachers who guided students in this process used the scaffolding method when students had difficulties, and also observed the students on a weekly-basis during the coding process and filled an observation form for each student.

4.3 Data Collection Tools

The data were collected from different data sources as a requirement of the case study method. In this way the findings could be analysed in detail. The data of the study were collected from four different data sources. The Figure presented below visualizes the data that were collected via data collection tools.

4.4 Interviews with Teachers

One-to-one interviews, that are commonly used in educational research studies (Creswell, 2012), were employed in this study to conduct interviews with teachers. The interviews were performed in the framework the semi-structured interview form of questions developed by the researchers. More detailed data can be collected in semi-structured interviews through spontaneous questions which provides flexibility to researchers (Creswell, 2012). The questions provided in the semi-structured interview form were formed based on the 'computational thinking evaluation framework' that was developed by Brennan and Resnick (2012). In addition to this framework, questions that aimed to reveal the changes that occurred in the 21st century skills that students had were included in the form. Moreover, the spontaneous questions were used to obtain more data. In this context, the following interview questions were addressed to the teachers during the interviews:

- How did students begin the game developing process?
- What kind of path did they follow to develop their games?
- How did students identify the data that can be useful for them while transferring the data to their games? (Abstraction)
- Did students program the games in modules? (Decomposition)
- Did students recognize the dysfunctional points in game codes on their own? What kind of solutions did they find? (Evaluation – Testing)
- Were students capable of recognizing the dysfunctional points in game codes and solving the problem? (Generalisation)
- Which skills of students have been improved as a result of coding their own games? (21st Century Skills)

4.5 Focus group interviews with students

Focus group interview is a method that can be used to receive the common views of a specific group on a case or event. Focus group interview is considered as a data collection method that is effective in the case that the interaction among the interviewees is high and when this interaction is expected to create more data (Creswell, 2012), also in the case that interviewees are working together on a specific task. Therefore, the interviews conducted with students were planned as a focus group interview.

4.6 Focus group interviews with pre-service ICT Teachers

Focus group interview method was used to receive the opinions of the pre-service Information Technologies teachers who were guided students while they were programming the games. Focus group interview method was chosen to reveal and compared the situation occurred in different groups given that pre-service teachers were performing the same task on different groups. The following questions were addressed to the pre-service teachers:

- How did students begin the game developing process? What kind of paths did they follow to develop their games?
    o How did students identify the data that can be useful for them while transferring the data to their games? (Abstraction)
    o Did students program the games in modules? (Decomposition)
    o Did students recognize the dysfunctional points in game codes on their own? What kind of solutions did they find? (Evaluation – Testing)
    o Were students capable of recognizing the dysfunctional points in game codes and solving the problem? (Generalisation)
- Which skills of students have been improved as a result of coding their own games? (21st Century Skills)
    o Do you think that the students can work in harmony with other students attending another school? (Collaboration)
    o How do you evaluate the students on creating ideas to design their own games?   (Creativity)

4.7 Observation forms filled by pre-service ICT Teachers

Computational Thinking Self-Efficacy Scale developed by Kukul and Karatas (2019) was used as the baseline to observe students within the frame of computational thinking in the programming process. The observation form aimed to observe six different behaviours. The items included in the observation form are presented in Table 1. Each behaviour had a column for explanation where observers could use to take notes about the behaviour. These explanations provided detailed information about the observed or non-observed behaviour. The behaviours included in the observation form were scored as 1 or 0 which referred to the situations of being present or absent to control the development of the behaviours, and students' average scores for the observation form were calculated. In this way, the impact of each week on the behaviours was aimed to be observed using average scores.

Table 1. Items of the Observation Form and Related Concepts

Observed Behaviour	Related Concept
The student is able to create the steps necessary for solving the problem.	Algorithmic Thinking
The student is able to identify the dysfunctional points in the steps specified for solving the problem.	Evaluation or Testing/Debugging
The student is able to break up the problem into sub-problems.	Decomposition
The student is able to use previous experiences to solve the problems.	Generalization
The student is able to overlook the unnecessary information/data while solving the problem and focus on the solution.	Abstraction
The student is able to visualize the data to be used for solving the problem or the solution itself.	Abstraction

4.8 Data Collection and Analysis Process

In the data collection phase, different data sources were employed due to the nature of the case study method. In

the first place, the contributions of game programming in computational thinking skills aimed to be identified through three different focus group interviews conducted with students. Each focus group interview took around 15 minutes and both students and their parents were informed about the objectives of the interviews. Students were also informed that they can opt-out from the interviews. After receiving the required consents, each interview was recorded.

Afterward, semi-structured interviews were conducted with the participation of the teachers who were in charge of the student clubs, who also had an opportunity to observe the students in different environments apart from the club activities. The interviews conducted with two teachers lasted for 70 minutes in total. Finally, a focus group interview was conducted with three pre-service Information Technologies teachers who observed the students during the game coding activity. The interview took around 35 minutes.

The interviews were analysed in three phases including: (1) organization of the data, (2) coding and reducing data (3) representing data with tables and discussion (Creswell, 2007). In the first phase, the data were transformed into text files from sound files and each participant was given a nickname. In the second phase, the data were separated into sections using the constant comparative method (Glaser, 1965) and the data sections were continuously compared according to their similarities and differences.   To name the data sections, both the framework developed by Brennan and Resnick (2012) and the concepts of computational thinking included in the literature were considered. In the analysis of the data, opinions of different data sources were compared, and themes were created. In the final phase, the codes and themes were presented in tables and discussed within their own contexts. As described by Creswell (2007), these three phases proceeded interactively in a spiral structure.

To ensure the credibility of the findings, the principles proposed by Lincoln and Guba (1985) were adopted. The data were also analysed by another researcher to avoid the prejudice of the researchers towards the research problem, and a consensus regarding the results was reached after the discussions with researchers based on negotiated coding approach developed by Garrison et al. (2006). The obtained codes and themes were explained detail within their own context and the transferability of the findings to other contexts was ensured. Furthermore, the data obtained from different data sources, apart from the interviews, were compared with the findings of the interviews. In this way, the findings were supported and discussed in detail.

In addition to the interviews, the pre-service teachers were asked to fill observation forms each week for students in their groups. The observation forms filled by the pre-service teachers were analysed and weekly behaviour changes were identified. Furthermore, observation notes of each pre-service teacher were used to support the interview data. Given that each pre-service teacher observed another student, no comparisons were made between the observation scores, and the improvement of each observation item was followed on a weekly basis.

5. Findings

5.1 Game Programming and CT

To identify the impact of game programming on students' CT skills, interviews were conducted with all relevant stakeholders.   The concepts concerning CT that were changed as a result of the interviews were presented in Table 2.

Table 2. Game Programming and CT Components

	According to the Teachers	According to the Pre-Service Teachers	According to the Students
Data collection	X	X	X
Abstraction	X	X	X
Testing - Debugging	X		X
Decomposition	X	X	X
Generalization	X	X	

The Table shows that teachers who were in charge of the clubs and pre-service teachers shared similar opinions. However, given that pre-service teachers observed different groups, there were differences regarding the impact

on students' CT skills. For example, students in a group attended the club activities two years in a row, and almost all of the changes regarding the CT components were present for this group. However, this change was not observed for students on another group.

In analysis of the CT skills, based on the paths that were chosen by the children while programming their games, the teachers stated that the students began to the process by collecting data.

> *"We define the theme of the game. Afterward, the research process about the theme takes place. We turn on the projector in the laboratory together. We make a detailed research about the theme. At the end of the day, we give tasks to students about the theme. Everyone make research on a specific topic. In our next meeting, they present their research" (Teacher 1).*

One of the pre-service teachers shared a similar opinion and stated that students began the process with collecting data.

> *"The topic was about the space and therefore we needed to collect primary data. At first, we collected data about space." (Pre-Service Teacher 3)*

Students also stated that most of the groups began the process with data collection. The interviews conducted with students showed that teachers' guidance towards making research was effective. On the other hand, the results of the interviews showed that some of the groups avoided the guidance and did not make any research.

> *"At first, we made computer-based research to decide where our work will take place." (Student 2)*

> *" We started by creating the background. Afterward, we decided on the character and started to write the codes" (Student 4)*

However, it drew attention that students who began the process with making research were highly interested in the selected theme for the game, which was 'Space'. The expressions of the same pre-service teacher revealed this situation clearly:

> *"Some of these children are very interested in space. When they had a chance to design a game related to space, they felt the need of making more research and to acquire more information. They read magazines and read about space. They even watch the movie called Interstellar. In the end, they were asking about going there, what they can see in Titan, and if they can make different research projects?" (Pre-Service Teacher 3)*

On the other hand, students in the other group were not interested in the theme 'space' and therefore, they had a motivation level at the beginning of the programming activities. This situation was expressed by a pre-service teacher as follows;

> *"… we imposed the space theme for the ones who are not interested in. In the end, we got games irrelevant to space. They had prejudices towards the selected theme. We created a Head Ball game, it caught students' attention and they all wanted to do it. They are more comfortable when they developed games related to themes they choose. They should be free to choose their themes. " (Pre-Service Teacher 3)*

About the problem in the selection of the theme, another pre-service teacher stated that;

> *"They were not familiar with the theme. The theme chosen by the teacher was not interesting for them and they had no information about it. The internet-based research was not enough at that point." (Pre-Service Teacher 1)*

Abstraction is among the key components of the CT components (J. Wing, 2010; Jeannette M Wing, 2008). Teachers' statements revealed that students were capable of making an abstraction.

> *"Although students did not take the data that can use from during the research process by saying 'I can use it later'', I saw that they were influenced by the research process while they were designing their games." (Teacher 1)*

The student expressed that they made abstraction in different dimensions.

> *"We take some parts of the research and used them in the game. We designed it on paper." (Student 2)*

> *"Our research helped us to figure out what kind of a world we want to design and the research process was useful for that." (Student 6)*

However, this situation was not valid for every student. One of the pre-service teachers expressed that;

> *"Even we have just gained the habit of talking about the planning phase of a project in detail. The fourth-grade students do not have this habit. They want to insert all objects in the program in the game nut they don't do it with a certain logic. For example, we made the planning on paper at first. We write down our thoughts on paper. " (Pre-Service Teacher 1)*

The teachers' expressions showed that students were in the phase of testing and error debugging processes related to their games. However, pre-service teachers did not mention such a skill. The interviews made with students showed that some of the students made debugging, and some of them could see the error yet did not understand the source of the mistake and spend hours for trial and error.

> *"After designing the game they combine their modules and they play the game for an outsider. They correct their games according to the feedback. As they design their own games, they are very aware of the track of the game. Therefore, they can correct their own mistakes. They know where the error is and they also know the reason for it better than others." (Teacher 2)*

> *"We placed the stones and we wanted to blow up all of them but they didn't. Then I wanted to see why they didn't blow up but we find anything. We put a lot of effort but then we found the reason and solved the problem. "(Student 3)*

> *"We had a difficult time while coding. When we faced a difficulty we discussed with the group members and solved the problem." (Student 8)*

> *"We made a lot of mistakes in the coding phase. We corrected them through the trial and error method." (Student 7)*

During the activities, students were working in groups and they had cooperation within the groups. This situation contributed to their skills to break the problem into sub-problems. The given situation was realized both by the pre-service and teachers. Some of the students also stated that they separated the tasks into different sections.

> *"Of course they break the game into pieces. Everyone is working on a section. In this way, they could show their talents better and easily in their own fields. Afterward, they presented their results to their friends. They worked in a way that 2-3 students could design a game." (Teacher 1)*

> *"Each student got a task and started to work" (Student 2)*

> *"...we gave students individual tasks. They had to complete 3 sections." (Pre-Service Teacher 3)*

The teachers stated that students' generalization skills were improved through game programming activities.

> *"They can absolutely develop ideas. They can express opinions like this is missing here or this code is absent, so we need to add it. They can absolutely generalize what they have learned by considering other problems. The children are very good at using KoduGame. Therefore, I don't think that they will have difficulties if they use scratch" (Teacher 2)*

Pre-service teachers believed that the given situation is all about having experience. They expressed that the group that was joining programming club activities previously has enabled them to generalize their knowledge to new learning situations and programming problems very easily. On the other hand, the students in the other group did not have such kind of generalization skills. Given that the theme did not draw the attention of the students in terms of improving generalization skills, it was believed that students would not be capable of doing it. However, related to the generalization skills of students, the advisor pre-service teacher of the experienced group said that;

> *"4th and 5th-grade students were saying 'we can handle this' and they could even code the parts which I found difficult. They started to code directly." (Pre-Service Teacher 3)*

5.2 Observation of CT

Figure 1 shows the improvements occurred in students' skills based on the average scores calculated for each item in the observation form on weekly basis according to the observations made by pre-service teachers.
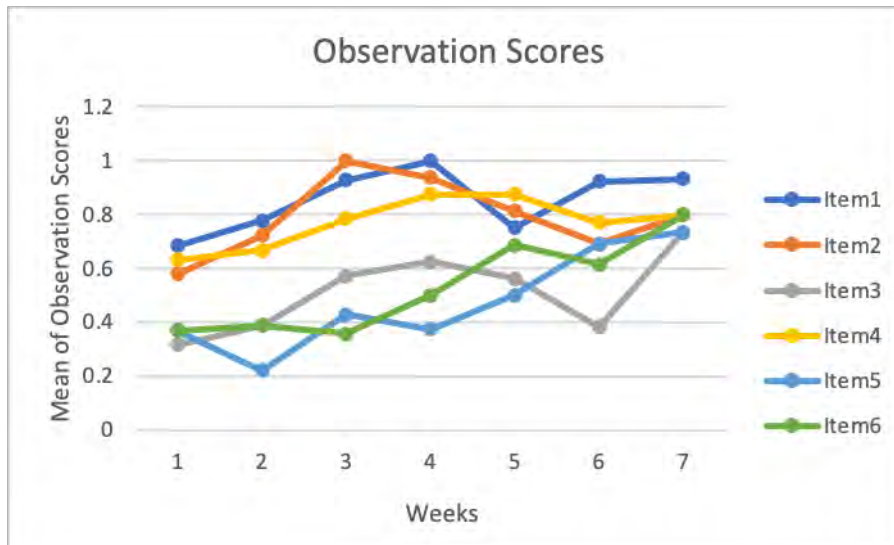
Figure 1. Observation Scores

The Figure shows that skills that were observed at the lowest level starting from week one were students' abstraction and decomposition skills. The results showed that other skills could be observed more frequently. Considering the whole CT skills, it is seen that students' CT skills showed an increase in scaffolding-based activities. These results support the interviews made with the stakeholders. Furthermore, although not all of these behaviours showed a regular increase, it is possible to say that there were a difference and improvement in all behaviours between the first and the last week.

The separate analysis of these behaviours showed that students' algorithmic thinking skills were increased between the first and the last week. However, it was the most frequently observed behaviour in students at the beginner level, and therefore the increase was relatively lower in the last week. In this case, the students with programming experience might have an impact. Furthermore, students' algorithmic thinking skills might be improved in different courses and therefore the level of students was found high at the beginner level.

Evaluation or Testing/Debugging item showed that there was an increase as in the case of algorithmic thinking, however, the beginner level was high. It can be argued that the scaffolding activities performed by the pre-service teachers were effective for students to debug the errors and make evaluations.

The analysis of the third item showed that decomposition skill was the least observed skill in the first week. This situation can be explained by the fact that students did not face such kind of problems before. In addition, the lack of problem-solving skills might play a role in this situation. As time progressed, the skill also had been improved and the given situation can be explained with the impact of the scaffolding activities on decomposition skills as mentioned by Rogoff (1990).

The results also showed that there were an improvement in students' generalization skills, however, the increase observed in this skill in the first week was relatively lower. The skill of students that showed the highest improvement is abstraction skill. From this aspect, it can be argued that scaffolding-based game programming activities help to improve abstraction skill. The abstraction skill, which was observed at considerably low levels, in the beginning, caught up with other skill levels in the last week. Given that the abstraction skill was among the key skills for CT (Jeannette M Wing, 2008), it can be argued that scaffolding-based game programming activities made a contribution to the improvement of CT.

5.3 Game Programming and 21st Century Skills

The study aimed to examine whether game programming had an impact on students' 21st-century skills, besides their CT skills. At this juncture, the 21st Century Skills that was specified by International Society for Technology in Education was used as a framework. The study aimed to identify which skills of students that were presented in the framework showed an improvement in this process.

Table 3. Game Programming and 21st Century Skills

	According to the Teachers	According to the Pre-Service Teachers	According to the Students
Innovative Designer	X	X	X
Knowledge Constructor	X	X	
Creative Communicator	X		X
Global Collaborator	X	X	X
Computational Thinker	X	X	X
Other Variables			
Motivation	X	X	
Self-Confidence	X		X

The teachers expressed that many 21st Century Skills of students who joined the club activities were improved. The teachers also stated that students were able to come up with creative ideas. They expressed their opinions as follows;

> "...when were with them in the coding process they design the space and they create a story within that. There are 16 children and they work on STEM on Wednesday and on coding on Friday. There is a considerable improvement in their creativity" (Teacher 1)

On the other hand, pre-service teachers did not describe the situation from a very positive perspective. The main reason for this situation is the fact that the theme of the games which will be designed by the students was chosen by a teacher. This situation might hamper students to produce original outcomes. Furthermore, a pre-service teacher argued that there is a gender-based difference in game programming activities. According to the pre-service teacher, girls produced fewer creative ideas compared to boys.

> "Except one group, students could not produce original outcomes. I think the inability of the teacher who was in charge of the club to plan the process played a role in this process. " (Pre-Service Teacher 1)

> "There is a game which is already existed, our students wanted to program the same game." (Pre-Service Teacher 2)

> "Three of our students were girls. I think that girls are less capable of thinking of games and creating them. They think very simply, probably I am also like that. Boys play more games; maybe too often, and therefore they are better in that. Not in the programming phase, but I think there is a gender-based difference specifically in creating games. Boys are really successful in creating them. Girls generally play with dolls so they think more simply while creating games. They combine two-three objects and then they leave it. " (Pre-Service Teacher 1)

The interviews made with the students showed that their opinions were similar to the opinions of the pre-service teachers. The students who gained experience in the previous year emphasized that their creativity was enhanced. However, this situation cannot be generalized to all students.

> "My creativity has been widened. I started to help people. For example, I thought about an elderly person who cannot walk and I wanted to design a tool to help that person to walk. " (Student 2)

> "I have become more creative. Do you know the game Ben10? While I was watching it I said myself 'I can do it' and I programmed the game. Before the club activities I was not thinking about designing games." (Student 1)

In the research and data collection process which students performed before programming their games they could structure the information required for their games. However, the groups who were interested in the context of the game had a more efficient research process as well. On the other hand, other groups also made a progress under the guidance of teachers and pre-service teachers.

"But now they can search for the things they are want to find. They created a YouTube channel. They share their outcome on their channel. " (Teacher 1)

In the game programming process, the teachers put the most emphasis on the positive impact on students' communication skills. Both teachers highlighted that students' communication skills were considerably improved in the process of developing games and also afterwards.

"There was a big difference in their relationship with their friends. It is very obvious in their conversations. Other teachers are very surprised to see them talking in different places. " (Teacher 2)

"We can say that children's language skills and expression skills were improved in the coding process. Because they communicate about their experiences. They make presentations and share with us." (Teacher 1)

According to teachers, students improved communication skills and their efforts in club activities help them to become global collaborators. Similar opinions were also shared by pre-service teachers. Even the students who were not used to work in groups could work with group awareness.

"There is cooperation and counselling. It helps students to be less individualistic. These children are not special children selected from their classrooms; they are voluntary children who came here randomly. " (Teacher 2)

"They could get rid of their selfish feelings. Now there is no I, but there is we. We are working for us." (Teacher 1)

"The students in my group can definitely work with students even they don't know them" (Pre-Service Teacher 3)

Students also expressed opinions that can support the observations of pre-service teachers and teachers and stated that group work was a positive experience for them. Given that students' perspectives on the positive aspects of the group work were essentially the same, they indicated different points about these positive aspects.

"It was better for us that we were just two. Because we came up with different ideas and opinions while designing the game." (Student 2)

"It is difficult to be alone, I prefer to be in a team. Because as a team we can make the task with the help of different opinions and in the end we can make a decision. But when we are alone, we have to do it with only one idea" (Student 1)

"I think it is better to do it in a group. Everyone comes up with a different idea and I think it is a nice thing. Everyone can have an idea and we realize them, sometimes it creates problems. But we combine the ideas and solve the problem. " (Student 9)

"We can add different ideas. Some points remain missing when you work alone. Our friends can let us know about what is missing." (Student 5)

"We all have different characteristics. We all have parts that we are good at. It is good to combine them."(Student 6)

As a result of the impacts on the CT components provided above, it can be also argued that the CT skill has been also improved as a 21st-century skill. In addition to these skills, the teachers particularly stressed the impacts on psychometric variables such as motivation and self-confidence. The opinions of the students supported the expressions of their teachers and they stated that there was an increase in their self-confidence levels.

"There is a burst of self-confidence. They are extremely happy. When they are happy they laugh, they gain self-confidence. We are exactly at this point. It happened with the help of the club activities but it reached a peak with the coding activities. They could produce an outcome with club activities but with the coding activities, they found a chance to show it to others. This led to a burst of self-confidence in students. " (Teacher 2)

"Their self-confidence levels have been increased when they started to crease something. Developing their own products is an important factor." (Teacher 1)

"They believed that someone can be happy to play the games they designed. When other like their games they are more motivated. They try to make a better game." (Pre-Service Teacher 3)

> *"I am more comfortable in group work now. Normally I am very shy but I was not embarrassed while we were presenting the game in the exhibition." (Student 8)*

The given statement of the student can be interpreted as a sign of self-confidence and it can be also considered as an impact of coding on communication skills.

6. Discussion

This research study aimed to analyse the impact of scaffolding-based game programming activity on students' CT and 21st century skills. According to the results of the study, it is seen that some psychometric variables can be affected during the game programming process. In this direction, the findings of the study are summarized in Figure 2.
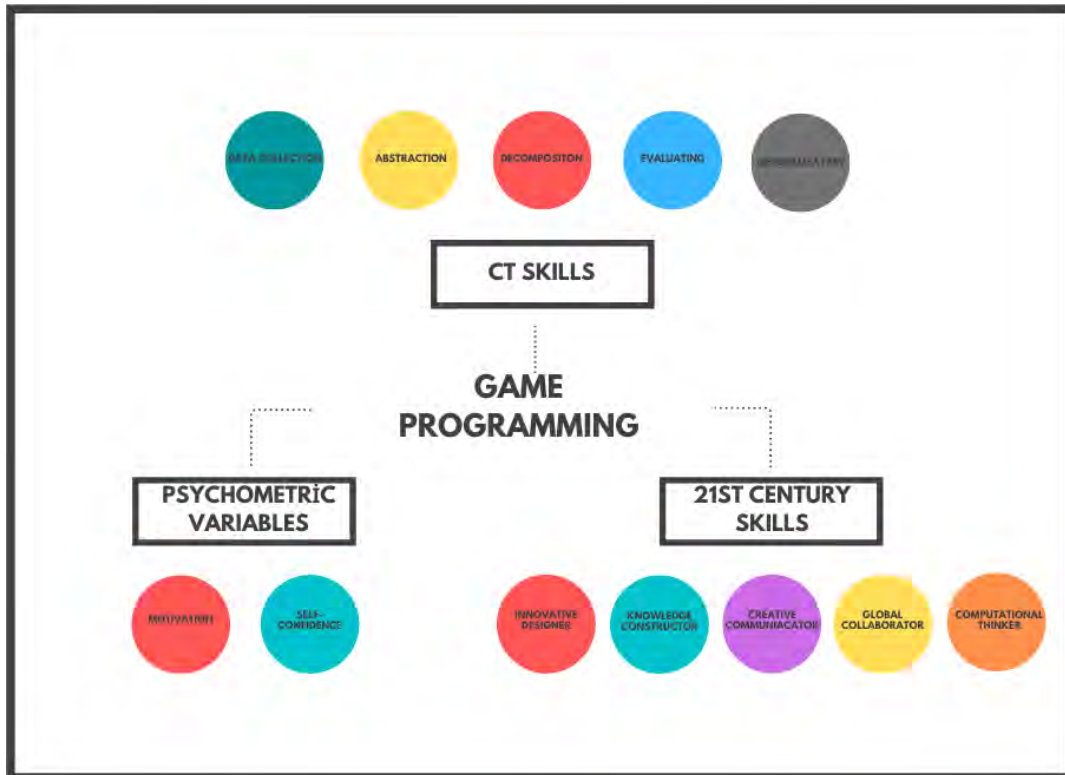

Figure   1. Summary of the Research Findings

The study findings revealed that students began the programming with the data collection process. This skill is also presented in the operational CT definition made by the Computer Science Teacher Association and International Society for Technology in Education (ISTE, 2011). According to the findings, scaffolding performed by pre-service teachers was effective in the data collection process. However, in the case that students are not interested in theme they need to make a research on, scaffolding performed by teachers may remain insufficient. However, it is stated in the literature that scaffolding can be more effective if performed by peers (Thomas, Ge, & Greene, 2011; Vygotsky, 1978). This situation can be investigated with different studies.

Individuals who can program computers are thought to have high level abstraction skills (Wing, 2006). It was tried to help students gain this skill in a shorter time with scaffolding. The scaffolding performed by the pre-service teachers was effective in improving students' abstraction skills. This finding coincides with the findings of the study (Feng, & Chen, 2014), which revealed that programming with scaffolding is better understood and thus contributes to the development of abstraction skill. However, the expressions of the pre-service teachers showed that they had gained this skill very late. Furthermore, the programming experience of the pre-service IT teachers might also lead to an improvement in students' abstraction skills. Therefore, if scaffolding was performed by individuals with low abstraction skills rather than pre-service IT teachers, it might make a negative impact on students.

The research findings showed that students (particularly the ones with programming experience) had generalization skills. However, the increase in the generalization skills was found low, and this situation was

associated with the insufficient levels of scaffolding performed by the pre-service teachers. The literature shows that without scaffolding, students face problems in transferring their previous experiences in solving new problems (Lye & Koh, 2014), and they tend towards the trial and error method (Biesta & Burbules, 2003). The same situation is also relevant in the case of debugging, and it was seen that students tried to debug the error instead of trying to find the source of it.

The findings related to the 21st-century skills of the students showed that all stakeholders emphasized the increase occurred in students' innovative designer, global collaborator and computational thinker skills with the game programming process. It is possible to see similar studies in the literature (e.g., Bermingham et al., 2013; Zimmerman, 2007). In addition, it was seen that students had become knowledge constructors and creative communicators, although not all stakeholders shared the same opinion. It seems that game programming attracts students' attention and helps them communicate with each other (Thomas, Ge, & Greene, 2011). The fact that the theme of the game designed by the students was chosen by the teachers, which did not draw their attention, might prevent them from becoming knowledge constructors.

The study findings showed that students' motivation and self-confidence levels were increased after the programming activities and presenting their outcomes. This finding is in parallel with the studies in the literature (e.g., Vos et al., 2011; Bermingham et al., 2013; Garneli, & Chorianopoulos, 2018). The students' designing their own games allowed them to see that they could achieve something. This situation also affects the belief that they can achieve different things in different areas (Guay, Boggiano, & Vallerand, 2001). At this point, teachers should take into account that self-confidence will not increase in all students. For this reason, teachers should support all individuals to increase their self-confidence.

7. Conclusion

One of the limitations of this study was related to the fact that there was no instruction design regarding the club activities performed by the students. In this way, the impact of student club activities on students' CT and 21st-century skills were analyzed. By doing so, the study aimed to make recommendations for teachers who deliver the courses and who are in charge of the student clubs on the implementation process. In future studies and implementations, educators may support their instructions of programming through different scaffolding strategies. Furthermore, experimental studies can be conducted to identify the impacts of different strategies. Teaching programming via real-life problems that can attract students' attention may also enhance the improvement of CT skills. In addition, it should be taken into consideration that students can become innovative designers with content that they find interesting.

Conducting research studies to analyse CT skills of teachers who will perform scaffolding for students, and implementations to improve teachers CT skills may assist to perform a more appropriate scaffolding. In this way, more effective results can be received from students' CT skills.

Considering that students' generalization skills are related to their experiences, designing instruction processes that will enhance their experiences may be useful. Enabling students to face more problems will increase their programming skills and this situation may also improve their CT skills.

The results of the study revealed that students were content with working in collaboration. Educators should take this point into consideration in their instruction designs. In fact, instruction designs that will allow using collaborative work in different phases of instruction, such as evaluation, can be planned.

8. References

Akcaoglu, M., & Green, L. S. (2019). Teaching systems thinking through game design. *Educational Technology Research and Development*, *67*(1), 1-19.

Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International journal of child-computer interaction*, *19*, 30-55.

Bean, T. W., & Patel Stevens, L. (2002). Scaffolding reflection for pre-service and inservice teachers. *Reflective Practice, 3*(2), 205–218.

Bermingham, S., Charlier, N., Dagnino, F., Duggan, J., Earp, J., Kiili, K., ... & Whitton, N. (2013). Approaches to collaborative game-making for fostering 21st century skills. In *European Conference on Games Based Learning* (p. 45). Academic Conferences International Limited.

Biesta, G. J. J., & Burbules, N. C. (2003). Pragmatism and educational research. Lanham, MD: Rowman & Littlefield.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking.* Paper presented at the American Educational Research Association, Vancouver, Canada.

Chiazzese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Engaging primary school children in computational thinking: Designing and developing videogames. *Education in the Knowledge Society*, *19*(2), 63-81.

Creswell, J. W. (2012). *Educational Research Planning, Conducting, and Evaluating Quantitative and Qualitative Research* (4th Edition ed.). Boston: Pierson.

Dennen, V. P. (2004). Cognitive Apprenticeship In Educational Practice: Research On Scaffolding, Modeling, Mentoring, And Coaching As Instructional Strategi es. In D. H. Jonassen (Ed.), *Handbook of Research on Educational Communications and Technology* (Second Edition ed.). New Jersey: LAWRENCE ERLBAUM ASSOCIATES.

Denner, J., Campe, S., & Werner, L. (2019). Does computer game design and programming benefit children? A meta-synthesis of research. *ACM Transactions on Computing Education (TOCE), 19*(3), 1-35.

Dolgopolovas, V., Jevsikova, T., Savulionienė, L., & Dagienė, V. (2015). *On Evaluation of Computational Thinking of Software Engineering Novice Students*.

Feng, C. Y., & Chen, M. P. (2014). The effects of goal specificity and scaffolding on programming performance and self-regulation in game design. *British Journal of Educational Technology, 45*(2), 285-302.

Fowler, A., & Khosmood, F. (2018). The Potential of Young Learners Making Games: An Exploratory Study. In *2018 IEEE Games, Entertainment, Media Conference (GEM)* (pp. 1-9). IEEE.

Garneli, V., & Chorianopoulos, K. (2018). Programming video games and simulations in science education: exploring computational thinking through code analysis. *Interactive Learning Environments, 26*(3), 386-401.

Ge, X., & Land, S. M. (2003). Scaffolding students' problem-solving processes in an ill-structured task using question prompts and peer interactions. *Educational technology research and development, 51*(1), 21-38.

Genç, Z., & Karakuş, S. (2011). Learning by Design: Using Scratch in the Design of Educational Computer Games. 5th International Computer & Instructional Technologies Symposium (ICITS), Elazığ, Turkey. Retrieved From http://goo.gl/KX5Psz

Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE* (Vol. 2007).

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher, 42*(1), 38-43.

Guay, F., Boggiano, A. K., & Vallerand, R. J. (2001). Autonomy support, intrinsic motivation, and perceived competence: Conceptual and empirical linkages. *Personality and Social Psychology Bulletin*, *27*(6), 643-650.

Hainey, T., Baxter, G., & Ford, A. (2019). An evaluation of the introduction of games-based construction learning in upper primary education using a developed game codification scheme for scratch. *Journal of Applied Research in Higher Education, 12* (3), 377-402.

International Society of Technology in Education [ISTE] (2011). *Computational Thinking Leadership Toolkit*. Retrieved from https://cdn.iste.org/www-root/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4

International Society of Technology in Education [ISTE]. (2016). 21st Century Standarts for Students. Retrieved from https://www.iste.org/standards/for-students

Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing, 4*(3), 583.

Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, *21*(4), 371-391.

Kukul, V., & Karatas, S. (2019). Computational Thinking Self-Efficacy Scale: Development, Validity and Reliability. *Informatics in Education, 18*(1), 151-164.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., . . . Werner, L. (2011). Computational Thinking for Youth in Practice. *ACM Inroads, 2*(1), 32-37.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51-61.

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). *Computational Thinking in K-9 Education.* In Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference (pp. 1-29)

McLoughlin, C. (2002). Learner support in distance and networked learning environments: Ten dimensions for successful design. *Distance Education*, *23*(2), 149-162.

Partnership for 21st Century Learning (2007). Framework for 21st Century Learning. Retrieved From:

http://static.battelleforkids.org/documents/p21/P21_framework_0816_2pgs.pdf

Perkovic, L., & Settle, A. (2010). Computational thinking across the curriculum: A conceptual framework. *College of Computing and Digital Media Technical Report*, 10-001.

Rogoff, B. (1990). *Apprenticeship in thinking: Cognitive development in social context*. New York: Oxford University Press

Román-González, M. (2015). *Computational Thinking Test: Design Guidelines And Content Validation*. In Proceedings of EDULEARN15 conference (pp. 2436-2444).

Roman-Gonzalez, M., Perez-Gonzalez, J.-C., & Jimenez-Fernandez, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678-691.

Roman-Gonzalez, M., Perez-Gonzalez, J.-C., Moreno-Leon, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior, 80*, 441-459.

Thomas, M. K., Ge, X., & Greene, B. A. (2011). Fostering 21st century skill development by engaging students in authentic game design projects in a high school computer programming class. *Journal of Educational Computing Research*, *44*(4), 391-408.

Van Laar, E., Van Deursen, A. J., Van Dijk, J. A., & De Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. *Computers in Human Behavior, 72*, 577-588.

Vos, N.H., van der Meijden, H. and Denessen, E. (2011), "Effects of constructing versus playing an educational game on student motivation and deep learning strategy use", *Computers & Education, 56* (1), 127-137.

Vygotsky, L. (1978). Mind in Society: Development of Higher Psychological Processes (Cole M., Jolm-Steiner V., Scribner S., & Souberman E., Eds.). Cambridge, Massachusetts; London, England: Harvard University Press. doi:10.2307/j.ctvjf9vz4

Yükseltürk, E., & Üçgül, M. (2018). Blok Tabanlı Programlama. In Y. Gülbahar & H. Karal (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi.* Ankara: Pegem Akademi.

Wang, L. C., & Chen, M. P. (2010). The effects of game strategy and preference-matching on flow experience and programming performance in game-based learning. *Innovations in Education and Teaching International*, *47*(1), 39-52.

Weinberg, A. E. (2013). *Computational Thinking: An Investigation Of The Existing Scholarship And Research.* [Unpublished doctoral dissertation]. Colorado State University,

Wing, J. M. (2006). Computational Thinking. *Communıcatıons of the ACM, 49*(3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725.

Wing, J. (2010). *Computational Thinking: What and Why?*
https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Zimmerman, E. (2007) Gaming Literacy: game design as a model for literacy in the 21st century. Harvard Interactive Media Review, *1*(1), 30-35.