



Teaching Block-Based Coding to a Student with Autism Spectrum Disorder

Canan Sola-Özgüç¹

Damla Altın²

Abstract

Introduction: The aim of this study is to review the process of teaching coding to a student with autism spectrum disorder (ASD) and to examine that student's educational gains, problems encountered during process, and suggestions for solutions.

Method: The research was designed as a case study and involved a three-month instruction process, during which the instruction sessions were prepared via the explicit instruction method of coding skills for a 10-year-old boy with autism spectrum disorder. Further, the contributions of this study were analysed through the data obtained via content analysis method with the NVivo 10 program.

Findings: The findings indicated that the participant demonstrated significant gains as a result of the coding teaching process. He successfully completed the first and second levels in the Code.org platform by reducing the level of help given under the researcher's supervision. The participant completed the last two lessons of his second level without any help and became independent in coding skills. The gains increased the participant's motivation; he further stated that he wanted to make coding his future profession.

Discussion: The participant not only gained skills in programming, he also learned new skills, e.g., finding the document he saved on a computer, taking a screenshot, and printing something out. According to the literature, programming teaching not only contributes to career planning in the field of engineering, these skills can also be used in daily life or will be needed in other professions. It has been emphasized that individuals with ASD should not only use technology as consumers, but also should be productive individuals in the field of technology by developing their computational thinking skills.

Keywords: STEM, autism spectrum disorder, coding, code.org, computational thinking, case study.

To cite: Sola-Özgüç, C., & Altın, D. (2022). Teaching block-based coding to a student with autism spectrum disorder. *Ankara University Faculty of Educational Sciences Journal of Special Education*, 23(3), 565-594. <https://doi.org/10.21565/ozelegitimdergisi.822554>

¹**Corresponding Author:** Assist. Prof., Sakarya University, E-mail: csola@sakarya.edu.tr, <https://orcid.org/0000-0002-4716-9028>

²Res. Assist., Sakarya University, E-mail: altin@sakarya.edu.tr, <https://orcid.org/0000-0001-8091-7625>

Introduction

With the rise of digitalization, today's students easily adapt to using technology. They tend to become manufacturers in the field as well, along with having little difficulty acquiring computer skills. In other words, people are now expected to be not only consumers but also producers (Ehsan et al., 2018; Kalelioğlu, 2015). For this purpose, science, technology, engineering, and mathematics (STEM) implementation is included in today's educational training programs. STEM aims to solve daily life problems for individuals of all ages from preschool to higher education while they prepare for careers in these STEM fields (Hwang & Taylor, 2016; Thomas, 2014). Over the last 20 years, employment opportunities provided by STEM have increased in addition to the increase of STEM-based training programs. Thus, interest in STEM-based job opportunities and STEM training programs has increased (Ehsan et al., 2018). It is emphasized there is a need for workforce in all areas of STEM; however, only 30% of this workforce have been provided by 2020; as such, these needs can be met by increasing the number of students studying in this field (Israel, Wherfel et al., 2015).

The necessity of including STEM applications into training individuals with special needs, STEM applications for individuals with special needs, and limitations encountered during the application of STEM to individuals with special needs are among the research topics (James et al., 2019; Taylor, 2018). Although education systems typically aim to provide equal learning opportunities to each student in STEM applications, students with special needs typically perform lower than children with typical development in these applications. This situation negatively affects the academic and career development of individuals with special needs (Burgstahler et al., 2012). For this reason, it is important for individuals with special needs to participate in STEM applications, so that they can contribute to their future career plans and become not only consumers but producers as well (AccessSTEM, 2007). According to the National Center for Special Education Research (NCSE) in 2011, less than 4% of individuals with special needs work in STEM-related jobs.

It is anticipated that a workforce of 5.5 million will be needed for jobs in the STEM field by 2022 (Newman et al., 2011; Vilorio, 2014). Because of a high-demand STEM field workforce in the future, it is essential that individuals with special needs can access STEM education as much as individuals with typical development to be employed in science, technology, engineering, and mathematics. Educators are introduced with new standards by STEM. This leads to the need for the access of individuals with special needs to STEM applications. For this purpose, new education programs, including STEM, are planned. Various arrangements are made regarding access to the computer field of STEM (Basham & Marino, 2013; Lindsay & Hounsell, 2017; Ludi & Reichlmayr, 2011). STEM applications with K-12 computer science framework were prepared in a comprehensive way for all students, including students with special needs. Computer Science Teachers Associations (CSTA) aim for students from all levels to have access to computer programming skills (Taylor et al., 2017). It is expected that these skills will help individuals transfer acquired skills to various areas in life as well as have careers in computational thinking. Computing skills given in K-12 instruction have shown that they develop problem-solving, collaborative research skills, form a basis for higher-order thinking skills, develop problem-solving in collaboration, and adopt positive attitudes about computer science. The students in this field also have fun while learning (Havlaskova et al., 2019; Kafai & Burke, 2014; Kalelioğlu, 2015). Considering these benefits, computational experiences will open career paths and provide a wide range of educational benefits for K-12 students who are typically-developing or have special needs.

Computational Thinking and Coding

In addition to basic knowledge and skills, having a high level of knowledge in technology is among the skills expected from individuals in the twenty-first century (Geist, 2016; National Education Association [NEA], 2012). Computational thinking skills, which are expressed as twenty-first-century skills, must be gained by integrating them into curriculums (Kong et al., 2020). Although studies emphasize its importance, research shows that teachers are not fully aware of computational experiences yet and their knowledge about computational thinking is limited (Israel, Wherfel et al., 2015; Kong et al., 2020; Mason & Rich, 2019). Computational thinking was defined by International Society for Technology in Education (ISTE; 2011, p. 1):

“As a problem-solving skill which is to organize and analyze the data in a particular logic when faced with problems; to make abstract data with models and simulations; to automate solutions with algorithmic thinking; to find the possible solutions to be able to use the resources effectively and efficiently and to use computers and other tools to generalize and transfer the solution process to the solution of wider problems.”

As computational thinking is basically defined as problem-solving skills, it is believed that students with special needs will have difficulty in complex problem-solving and abstract reasoning. Therefore, teachers who work with individuals with special needs should consider how they can support their students under these circumstances. Thus, students with special needs will also benefit from computing education (Israel, Pearson et al., 2015; Taylor, 2018).

Programming or coding with another name is the application and development process with various sets of commands to solve problems, enable human-computer interaction, and perform specific tasks via computers. Nowadays, coding is seen as a basic skill that every student should possess (Kafai & Burke, 2014). The literature states that teaching coding at an early age improves children's skills, e.g., analytical thinking, creativity, and problem-solving, and also has benefits for children such as producing problem-based solutions and providing collaboration skills (Kalelioğlu, 2015; Lindsay & Hounsell, 2017; Meccawy, 2017; Taylor, 2018; Vahldick et al., 2014). Basic algorithm knowledge and skills are taught through block-based coding platforms in coding education, which is prepared for early-age students. Students can develop these skills on platforms such as Scratch, Code.org, Kodable, The Foos, Tynker.

Coding in Special Education

Coding skills are taught to typically-developing students from different ages via different platforms. Further, improvements in students' computational thinking skills are obtained after this education (Geist, 2016; Papadakis et al., 2016; Ratcliff & Anderson, 2011; Sullivan et al., 2017). Studies in the case of students with special needs are limited. Limited studies indicated that individuals with intellectual disabilities and students with autism spectrum disorder (ASD) with limited cognitive skills could also acquire these skills (Knight, Wright & Defreese, 2019; Knight, Wright, Wilson et al., 2019; Taylor et al., 2017; Taylor, 2018). Moreover, coding skills contribute to the development of social skills and learning motivation of students with ASD. They transfer their coding skills and problem-solving skills they have acquired into their daily lives (Gribble, 2017; Knight, Wright, Wilson et al., 2019). It can be said that coding will have an essential place in the education and employment of students with ASD since individuals with ASD are strong in visual processes, focus on details, and have excellent skills in ensuring continuity in a subject (Samson et al., 2012). When the field preferences of individuals with ASD are examined, it is observed that they are interested in computer science. Compared to other disability areas, individuals with ASD prefer computer science the most with 16.22%. It is predicted that more individuals with ASD will prefer computer science in the future (Shattuck et al., 2014; Wei et al., 2013). Aware of this potential, large technology companies also develop appropriate employment programs to benefit from the characteristics of individuals with ASD such as being able to notice small details, go beyond the molds, and work meticulously, and support individuals with ASD in their work environments (Eiselt & Carter, 2018; Knight, Wright, Wilson et al., 2019). Considering these interests of individuals with ASD, it is vital for individuals with ASD to receive computer science education at an early age in terms of being directed to a professional field within the scope of computer science (Knight, Wright & Defreese, 2019).

Explicit instruction is a systematic and scientifically based application for individuals with special needs. Israel, Wherfel et al. (2015) state that explicit instruction is also used effectively in teaching computing skills. The constructionist approach is generally used more in coding or programming teaching. However, Taylor (2018), who thinks that this approach will not be the best choice for students with intellectual disabilities, used the explicit teaching method to first-grade students with intellectual disabilities in teaching basic skills in coding and stated that it was effective. Detecting errors and placing code are essential components of programming. The explicit teaching method can allow for more needed exercise, giving the student the chance to practice independently for the student to master. What is more, with explicit instruction, students' frustrations will be minimized in computational tasks owing to the brief explanations and monitoring, enabling the students to have a good command of each step with explicit instruction.

Israel, Pearson et al. (2015) examined the teaching process of school-wide computational thinking skills in which individuals with disadvantages and special needs were among the participants, aiming to support school-wide computational thinking skills. Teachers stated they did not know how to support students with cognitive disabilities while teaching them computational thinking skills. Taylor (2018) provided computer-programming education for three elementary school first-grade students with intellectual disabilities. These students were successful in programming via explicit instruction method but had limited gains in generalizing it to tablets. Taylor et al. (2017) revealed that participants were successful in delivering computer programming education to elementary students with Down's syndrome. The authors also suggested more research should be conducted on this subject. Ratcliff and Anderson (2011) also found that fourth-grade students with dyslexia, dyscalculia,

dysgraphia, and attention-deficit/hyperactivity disorder (ADHD) learned programming via LOGO. However, students with special needs were more successful when they were taught through the explicit instruction method and given certain tasks. Knight, Wright, Wilson et al. (2019) aimed to teach skills to three high school students with ASD, i.e., move, add light effects, create loops, and add sound effects to a robot via Ozoblockly (a block-based coding platform). All three students gained these skills by the end of the research.

As stated above, individuals with special needs should also be included in the instruction despite limited research and the need for information by teachers about the instruction of computational thinking to students with limited cognitive skills. This study aimed to demonstrate how a block-based coding was implemented for a student with ASD with mild intellectual disability by exploring the problems encountered and how they were resolved. In line with this goal, this study aimed to answer the following research questions:

1. How was the instruction process of block-based coding to a student with ASD?
2. What was the success rate of the student with ASD in block-based coding instruction?
3. What were the problems encountered during the instruction of block-based coding to the student with ASD? How was it resolved?
4. How did block-based coding instruction contribute to student with ASD?

Method

The aim of this study is to explain how the coding instruction on the Code.org site is carried out with a child with ASD and mild intellectual disability. For this purpose, this study was conducted using a case study design with a single participant. Creswell (2014) defines this design as an approach in which the researcher collects detailed and in-depth data from one or more sources. The case study method was chosen in this study to examine the learning process of coding in detail.

The current study started when a teacher asked the second researcher for help. The teacher stated that her student was interested in information technologies, but she could not contribute much in terms of course content. The teacher stated that the student received support education within the scope of inclusive education, but this service could not be efficiently used in accordance with his area of interest. For this reason, the teacher requested support from the second researcher working at the department of special education. The researchers aimed to teach the coding skills which have become important subjects with the advancement of today's technology.

Participant

The participant was a 10-year-old boy with ASD and mild intellectual disability. The student had been receiving full-time inclusive education in the fourth grade of a public primary school. In addition, he had been receiving support education two hours a week about academic skills from a government-sponsored educational institution. The general intelligence score obtained from student's Wechsler Intelligence Scale for Children IV (WISC-R IV) test was 74.

Based on the interview with the teacher and direct observations, it is seen that the student was in parallel with his peers' performance in academic skills. Communication with his peers is extremely limited, he mostly spends time alone in the school yard between classes. He leaves the game a few minutes after he is included in an established game and prefers to sit alone. He is sensitive to loud noises. He leaves the setting when there is loud noise. He has an intense interest in technological tools. Realizing this interest, the teacher allowed him to use a computer between the classes and asked him to make presentations of the courses they covered by using computer PowerPoint for two hours a week. However, it was stated by the teacher that she was not able to lead the situation consciously except providing this opportunity.

Although the student has an interest in technology, his access to technological devices is limited by his family members due to their concern that his academic performance would decline. The student has behavioural problems such as examining a technological device without permission, violating personal areas, moving in the class during the lesson, and having tantrums at high volume.

Setting and Materials

The research was carried out in the special education unit of a public university. The special education unit was an 80 m² class on the ground floor of a two-story building. Low technological tools were whiteboard,

whiteboard pen, paper and toothpicks, marshmallows, and playing cards, which were used for unplugged activities. Samsung tablet, HP desktop computer, and HP printer were used as high technology tools.

Platform Used: Code.org

Code.org is a nonprofit organization founded in 2013. It aims to help every student from every school to learn computer science. Code.org consists of computer science fundamentals, independent courses, and code hour courses. The content of these courses includes applications related to algorithms, loops, and functions. Besides these courses, there are lessons for teaching applications, animation, and creating games with JavaScript. Activities on Code.org were prepared for users between the ages of 4 and 18.

The computer science fundamentals covered in this study consist of four levels regarding age levels as Course 1 (four to six years), Course 2 (above six years old), Course 3, and Course 4. There are several courses for ages 18 to 22 at each level. While most of these courses include activities on Code.org, it also features unplugged activities related to the main topics. Content on Code.org is not limited to activities for only students; there are also lesson plans and training videos for teachers who will perform these activities with students.

Data Collection

The research data consists of observation and semi-structured interviews conducted with several data sources, which are described in Table 1.

Table 1

Data and Type of Data Source

Aim	Data collection method	Data source
Evaluation of prerequisite skills	Observation Checklist	Video recording
Taking detailed information about child	Semi-structured interviews with his father	Audio recording
Gathering detailed information about the child	Semi-structured interviews with teacher	Audio recording
Instruction process	Observation	Video recording
	Researcher diary	Text
	Evaluation meetings	Audio recording
	Artifacts	Portfolio

Researchers developed a checklist to identify the academic and computing skills required for the student. In this checklist, computer terms, mouse functions, keyboard functions, and information technology competence were examined; reading comprehension, concepts (right-left, bottom-top, etc.), numbers, geometry and academic skill level were also identified. These consisted of 97 items.

Semi-structured Interviews

Semi-structured interviews were held with the student's father and teacher. These recordings were audio recorded. The second researcher conducted interviews with the father, which lasted 62 minutes. The first researcher conducted interviews with teacher, which lasted 38 minutes.

Unstructured Interviews

Interviews were conducted with participant's father and the participant himself before beginning the instruction sessions or at the end of the instruction sessions. The interviews were audio-or video-recorded.

Observation

Observation constituted primary data source. The observation sessions were video-recorded with the duration of 14 hours and 27 minutes.

Researcher Diary

During the application process, both researchers kept diaries. At the end of this process, a total of 60 pages of word documents were obtained. This data was used as a primary source with observations since it followed a chronological order and contained field notes for application.

Evaluation Meetings

Researchers wrote in their diaries after the sessions; the next day, after watching the videos, they came together to evaluate these sessions and plan the next instruction session. These meetings were audio-recorded with a total duration of 102 minutes.

Artifacts

Electronic designs were created by the student during the session. Handmade designs in unplugged activities constituted the student's work samples.

Researchers

Two researchers conducted this study. Both researchers work as faculty members in a special education department of a public university. The researchers have experience in carrying out qualitative research. The first researcher's role included planning the instruction session, observing the application, recording the application process, and diary writing. The second researcher's role included determining the coding stages to teach, making instruction plans, planning and providing necessary materials, and conducting sessions and diary writing.

Research Process

The research process was carried out in two stages. In the first stage, prerequisite skills of the participant for coding instruction were evaluated; the application process was planned by obtaining data from the family and teacher. After planning the instruction, the application was carried out in the second stage, and data regarding application were collected by making decisions about student performance.

Analysis

Data analysis was carried out during and after applications. During this process, researchers guided the research in the context by examining daily practice. Analysis processes after the application consisted of preparation of the data for analysis.

Researcher diaries, transcripts of family, and teacher interviews were read and coded by both researchers. Video recordings were examined in detail and analysed. Audio-recording transcripts and researcher diaries were transferred to the NVivo 10 program for analysis. Themes noted during analysis were reported as answers to research questions.

Validity and Reliability

Regarding validity, expert opinions were received from three different experts in the field of special education and qualitative research methods during the preparation of the interview questions. Expert opinion was received from three field experts to determine the student's current level of performance in computer coding. Data triangulation was performed using various data sources (Brantlinger et al., 2005; Creswell, 2014; Gay et al., 2006; Miles et al., 2014).

As for the reliability, two researchers independently coded data and created categories, i.e., agreement and disagreement categories. After the categories were created by the first researcher, both researchers formed categories into themes and sub-themes in parallel with the research questions. The reliability was calculated with the following formula $\text{Reliability} = \frac{\text{Agreement}}{\text{Agreement} + \text{Disagreement}} \times 100$ as suggested by Miles and Huberman (1994) (as cited in Patton, 2002). The reliability with the second researcher was calculated as 98%.

Procedural reliability was calculated for teaching coding via explicit instruction. For this aim, the first author marked on the Procedural Reliability Form within which tasks in the instruction sessions were included. Procedural Reliability Form includes tasks which the researcher should conduct. For instance, in the first level of Code.org, in which the phase of explicit instruction should be conducted, level-guided applications should be applied. Before the training, two researchers planned, and reconsidered this plan during the training. The formula for calculating procedural reliability is the number of observed behaviors divided by number of planned behaviors multiplied by 100 (Tekin & Kircaali-Iftar, 2001). It was found that training sessions took place with procedural reliability of %94.

Research Ethics

During this research, ethical rules were taken into consideration. The family was informed about the research process to carry out this practice. The process of the whole research in written form was submitted to the

family; their approval was received with consent form which included research process information and parent approval to participate study. They were informed that they could withdraw from the research at any time. Participant and the family members and teachers' names were not included in the research text. The student's images were arranged and used within the ethical rules' framework. The approval from the research ethics committee was received (document number 61923333/050.99/ by The Social and Human Sciences Ethics Committee of Sakarya University on 07/03/2020).

Results

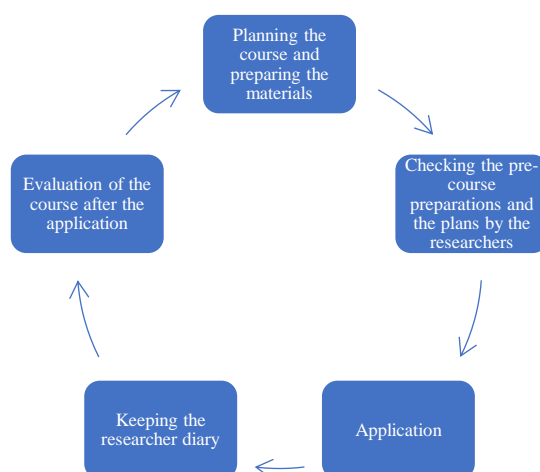
How was the Instruction Process of Block-Based Coding Presented to a Student with ASD?

Before the research application process, the researchers developed a checklist to determine whether the student met the prerequisite skills criteria for application (to teach on the Code.org platform). The computer skills were examined. During trial lessons with the student after the performance session, the researchers started teaching from Course 1, considering if the student can do Course 2, noting that, if he started from Course 1, he would better understand the programming logic.

In this study, the second researcher carried out coding instruction one-to-one on Code.org. Researchers further adopted the explicit instruction method as the teaching method for student. Accordingly, they planned the stages on Course 1 in Code.org as modelling, guided applications, and independent applications before starting the application processes, which were carried out two hours a day, as described in Figure 1.

Figure 1

Application Process



The researchers also determined which lessons to cover in the next session. Considering difficulty of the lessons and number of the activities in a lesson, it was decided how many Code.org lessons would be given in a session. The second researcher determined all lesson steps for the next session and planned them according to the explicit instruction method. Then, the two researchers checked planning and provided the necessary tools and equipment for unplugged activities and rehearsed them before each lesson. When the session started, the second researcher acted as the practitioner, and the first researcher acted as an observer. Video recordings of the application were watched by two researchers; they decided together what needed to be changed. They made necessary preparations for implementation of decisions for the next session. For example, the first four lessons in “Happy Maps, Play-Play, Puzzle and Maze” were covered in the first session, which lasted 30 minutes. Answers to the following questions were the primary purposes of this research: “How was the instruction process of coding to a student with ASD?” turned into questions “How long should be a course?”, “How many lessons should be taught during this period?” and “How do we shape the adapted method according to these stages?” With the increase in the given importance of coding instruction, it is thought that this data will make it easier to plan their lessons for researchers and teachers who want to use this platform for students with special needs in the future.

In the first lesson, the researcher became a model on the tablet computer and asked the student to apply the same step on a desktop computer. At the end of the first lesson, it was decided, via watching the video recordings, that the student did not need the modelling stage. Following the lessons, the number of stages of modelling was reduced. Guided practice and independent implementation stages were mainly included.

Course 1 was held in seven sessions; Course 2 was held in 11 sessions. The number of lessons in the sessions was planned, as the average lesson lasted between 40 to 50 minutes. Accordingly, a single lesson or several lessons were covered in some sessions. Details of Course 1 and Course 2 are given in Table 2.

Table 2

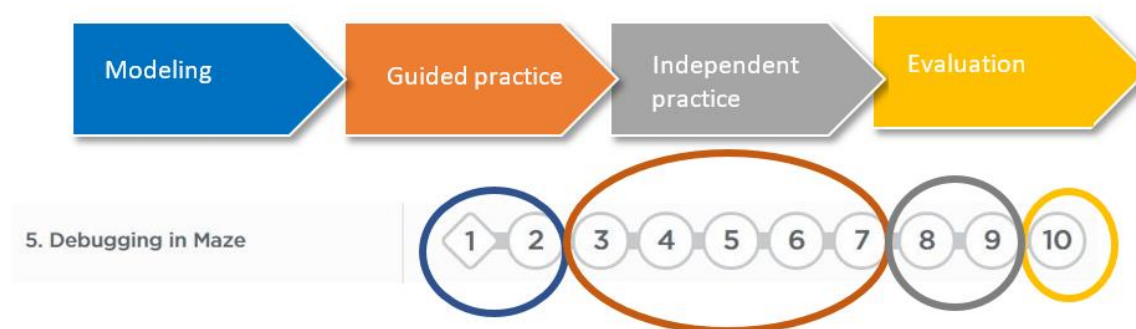
Courses Covered in Code.org Platform and Their Duration

Date	Course name	Total duration
03.01.2019	Course 1: Lesson 1, 2, 3, 4	00:31:18
03.04.2019	Course 1: Lesson 5, 6, 7	00:46:01
03.07.2019	Course 1: Lesson 8, 9, 10	00:44:38
03.11.2019	Course 1: Lesson 12, 13	00:27:02
03.14.2019	Course 1: Lesson 14	00:42:08
03.18.2019	Course 1: Lesson 15, 16	00:48:24
03.21.2019	Course 1: Lesson 17, 18	00:45:13
03.25.2019	Course 2: Lesson 1, 2, 3	01:11:24
03.28.2019	Course 2: Lesson 4	00:33:09
04.04.2019	Course 2: Lesson 5, 6	00:48:53
04.11.2019	Course 2: Lesson 7	00:10:37
04.15.2019	Course 2: Lesson 8	00:29:00
04.18.2019	Course 2: Lesson 9, 10	00:25:45
04.25.2019	Course 2: Lesson 11	00:38:19
05.02.2019	Course 2: Lesson 12, 13	00:48:33
05.06.2019	Course 2: Lesson 14, 15, 16	01:38:51
05.09.2019	Course 2: Lesson 17	01:18:24
05.13.2019	Course 2: Lesson 18, 19	01:21:55

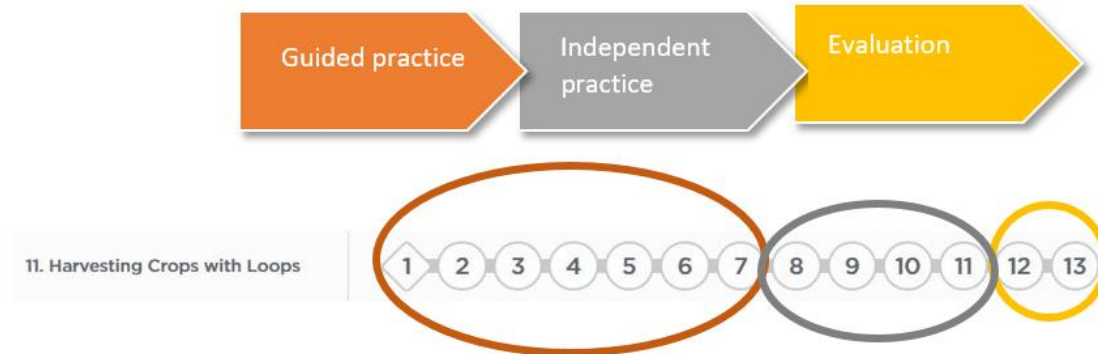
There are a number of practice steps varying between 8-14 in each course in Course 1 and Course 2. The second researcher conducted a total of seven lessons in the first two sessions by following all the steps of the direct teaching method. The researcher made the motivation step by explaining the subject to be learned at the beginning of each lesson. In the step of being a model, she showed the first two steps of each lesson on the tablet computer and asked the student to only watch herself during this time. Then, approximately 4 to 6 steps (varying by stages) were carried out with guided practices. In guided practices, the student performed the study from the computer, and the researcher showed the relevant step on the tablet computer with the model prompt whenever needed. After the guided practice, the two steps were recorded as the independent practices phase, followed by the evaluation step of the explicit instruction method, which was prepared as an evaluation in the last one and/or two steps of the platform.

Figure 2

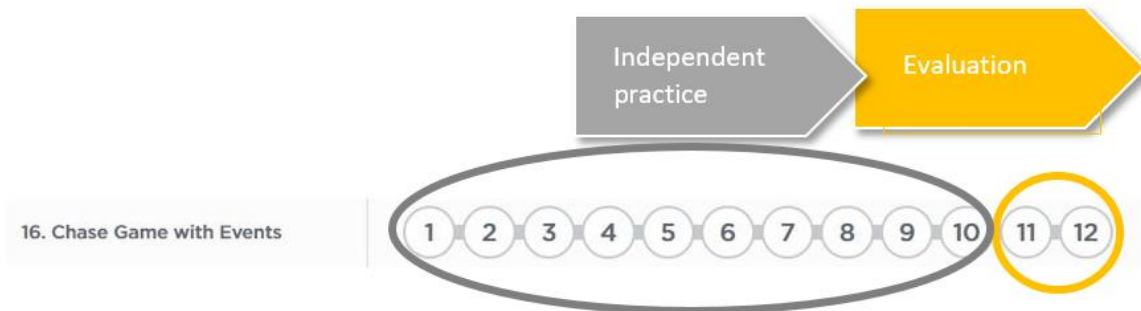
First Phase of Application



As a result of the student doing the stages without the need for prompts and being impatient while modeling on the researcher's tablet, it was decided that the stage of being a model in the reflection meeting was not necessary and that the teaching could be started from the guided practices phase in the sessions (The date of the reflection meeting was 02.03.2019). With this decision, 13 sessions were conducted with the guided practices phase, and the instruction was carried out. The process of the first phase of application is shown in Figure 2.

Figure 3*Second Phase of Application*

As the number of help received by the student from the practitioner decreased considerably and the design steps were passed, the independent practices step and the evaluation step were carried out in teaching (the date of the reflection meeting was 27.04.2019). He completed the requested tasks independently in three sessions (the date of the image recording was 09.05.2019). In this way, the prompts offered to the student was reduced by monitoring student performance. The student independently entered the platform, found the stage he was in, fulfilled the stages targeted for the session, and ended the course independently. It could be seen the second phase of explicit instruction in Figure 3 and the third phase of explicit instruction in Figure 4.

Figure 4*Third Phase of Application*

What is the Success Rate of a Student with ASD in Block-Based Coding Instruction?

As a result of this application, the student completed 342 levels and used 7372 code lines, as presented in Figure 5.

Figure 5*Performance of the Participant in the Application Process*

Kodlama Öğreniyorum		
Assigned to: Course 1		
Progress	Text Responses	Assessments/Surveys
Projects	Stats	Manage Students
Name	Completed Levels	Lines of Code
	342	7372

The student completed Course 1 and Course 2, i.e., 37 lessons in total. Lesson 11 in Course 1 could not be completed as it was an English puzzle activity called “Spelling Bee.” Lesson 17, which consisted of two stages

in Course 1, was partly completed as one stage was an unplugged activity. The lesson was completed in Course 1 and 2 by the student in Figure 6 and Figure 7, respectively.

Figure 6

Completed Lessons in Course 1 by the Student

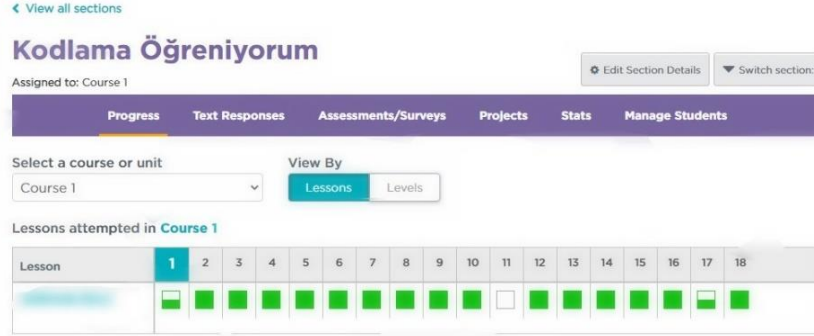


Figure 7

Completed Lessons in Course 2 by the Student



During the application, it was observed that the student did not get bored in Course 1 but made mistakes and found the mistakes himself in stages. In addition, the student supported the belief that starting from this stage was more constructive but not boring by saying, “I liked the Course 1 as well.” Starting from the first phase for students with special needs who does not have any experience in coding before makes them feel successful and also better establishes the basics of coding logic.

In the current study, in which prerequisite skills were evaluated, it was observed that the student had problems in the concept of direction in maze applications. As seen in Figure 8, bee should be coded as “turn right” on first corner. In this exercise, the student tried to write the codes according to his own location-not regarding character. After this subject, the student stood up and tried to find the position with hand and arm signs (Diary, p. 36, video recording, 04.04.2019). In this case, the researchers adopted the view that the student could not do it independently and at the desired level; thus, the researchers asked him to prepare worksheets under the guidance of his family. Progress was made in this area with family cooperation.

Figure 8

Maze Practice in Which the Participant Had Difficulty



The student spent most of the time in the design stages. Here, he created games, dialogues, and drawings by using the advancement, cycle, and condition skills that he gained at previous levels. Figure 9 shows an example of a drawing designed by the student.

Figure 9

A Design by the Participant on Code.org

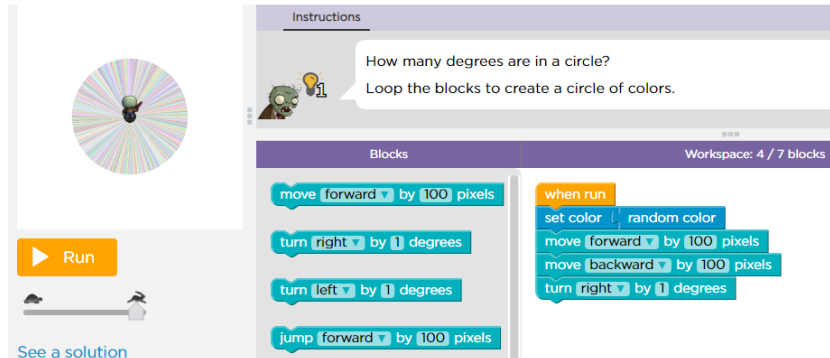


In this design, the student stated he had a lot of fun (video recording, 03.18.2019). When the researchers examined the work samples, they stated that dialogue written by the student may have reflected his inner voice. In other words, they thought the participant wanted to make friends and expressed this in his design.

The student's favourite activity was Play Lab: Create a Story; the activity that he disliked and was challenged most was Course 2: Lesson 11: Artist: Debugging, as shown in Figure 10. He had difficulty in stages such as calculating the circumference of the circle, i.e., giving the character instructions according to the circle angle. The student's fourth-grade curriculum did not include these subjects; only the concept of the circle was involved. This situation was conveyed to the Code.org platform, where it was stated by Code.org that the subject specified was included in the curriculum of students in that age group in the United States. In this case, the curriculum adopted by the country should be taken into consideration while working at this level of exercises at Code.org. The participant performed these activities via trial and error. Activities at this stage also contributed to the student learning the concept of the circle's circumference and angle.

Figure 10

Lesson in Which the Participant Had Difficulty



What were the Problems Encountered During the Instruction Process of Block-Based Coding to a Student with ASD? How Was It Resolved?

Under this title, three basic problems were found including student's behavioural problems, problems arising from the device used in coding instruction, and problems encountered in the use of Code.org platform.

The student's behavioural problems were encountered by the researchers. Although the student did not have the knowledge and ability to perform, he demonstrated behavioural problems such as trying to prevent the researcher in the teaching position to model, dominating the researcher's device, and not leaving the page when the course ended because of his intense interest in technology. The second researcher in the research diary stated: "Our lesson with student went well, but he interfered with me a lot, took my hand so that I could not stop him" (Diary, p. 7). Figure 11 shows participant's interference with the researcher.

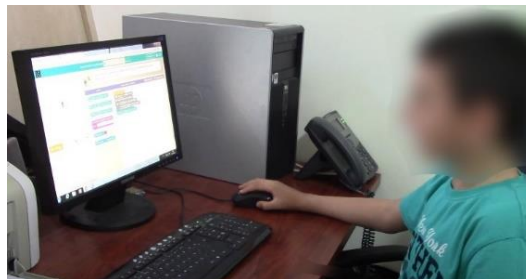
Figure 11*Participant's Interference with Researcher*

When the student's behavioural problems were persistent and there were disruptions during the lesson, the researchers set a number of rules in their evaluation meeting. Upon this decision, the following rules were set, and a behaviour contract was signed with the student, stating that they could practice if he followed the rules. The rules were as follows:

1. When I come to class, I greet my teachers and sit in my chair.
2. I wait for my teacher to turn on the computer.
3. When we start the lesson, I listen to my teacher, then I do the activity.
4. I do my activities on the computer.
5. I use the tablet with the teacher's permission.
6. During the activities, while one hand is working with the mouse, my other hand stays on the table.

The rules were implemented with the start of Course 2 (Diary, p. 33; image recording, 03.25.2019). In the first lesson, each rule was explained in detail, and reminders were made at the beginning of the lessons in the following sessions. Another problem encountered was the student's low motivation in unplugged activities. Although unplugged activities were held at another table, the student demonstrated behavioural problems such as constantly looking at the computer screen and touching the computer. The solution to this problem was to complete the unplugged activities at the beginning of each lesson when possible; if not possible, to turn off the monitor of the computer during the unplugged activity and to reinforce his interest in the unplugged activity.

Problems arising from the device used in coding instruction occurred when the student wanted to perform activities on the tablet-not on the computer. Performing coding on the tablet proved challenging for the student. Problems such as the use of two fingers on the tablet and touching another hand on the screen had consequences, e.g., not responding when clicked, entering the wrong tab. In addition, when a long code was written on the tablet screen, it could not be a whole, and it was only seen by scrolling the bar up or down (video recording, 03.07.2019). Researchers, upon encountering such problems, explained that the student was more successful on the desktop and, thus, was no longer using tablets in the research as seen in Figure 12.

Figure 12*Participant's Independent Activity*

Problems encountered in the Code.org platform included limitations such as stages that had English-based activities and the lack of translation for researchers. In preparations before lesson, screens were translated

to Turkish, and translations were written on post-it notes and attached to the screen. As some activities included puzzles in English, those lessons could not be covered in Turkish.

How Did Block-Based Coding Instruction Contribute to Student with ASD?

As a result of this practice, coding instruction had expected contributions for the student as well as the unexpected contributions. The student had significant achievements which were presented in Table 3.

Table 3

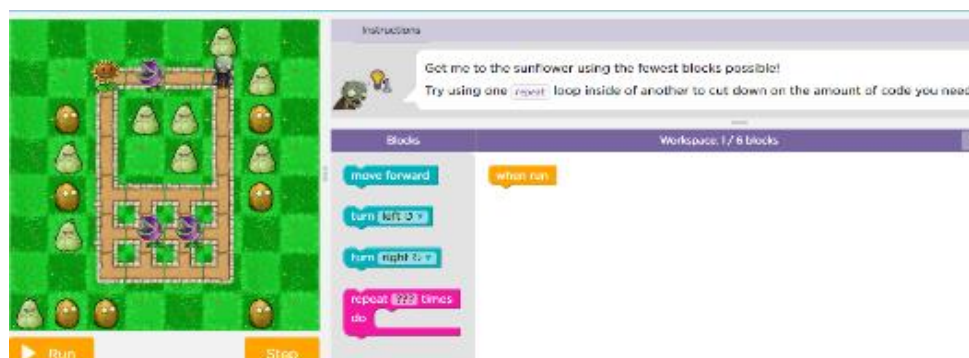
Contributions of Coding Instruction to the Student

Subthemes	Categories
Expected contributions of research	Improvement in problem-solving skills Improvement in self-management skills Motivation to produce
Unexpected contributions of research	Increase in vocabulary in English Improvement in computer skills (e.g., taking screenshots, printing) Improvement in social skills (e.g., greeting, initiating communication) Decrease in sound sensitivity

This study's expected contributions include improving problem-solving skills and self-management skills, and increasing the motivation to produce. In terms of improving problem-solving skills, the student gained the ability to develop a solution for a given problem by asking questions on how to get an object to the target in the shortest and fastest way at the end of the application, while he was only aiming to bring the object to target at the beginning of an application. As shown in Figure 13, to reach the sunflower, the character must go to a certain number of steps, turn, loop it, and not enter the sections with purple flowers during the process.

Figure 13

An Activity Improving Student's Problem-Solving Skills



Improving self-management skills is another contribution. The student showed progress in self-management with activities such as determining the code he made a mistake, extracting the wrong code, and trying again when he had incorrect code after dragging and dropping the codes. Also, thinking about a design for a long time, seeing his output by coding, publishing his work sample on Code.org, and reviewing it again in the next sessions showed his motivation, especially during lessons, including design.

The student can make future plans with support given in line with his interest. The student also expressed his opinions as "I will be an encoder; I will be a software engineer" during the process and at the end of the application. Study findings showed that students with ASD can make future plans and choose a field parallel to their interests as a professional career when their interests are supported.

One of unexpected contributions of the research was that he learned to translate texts into his own language using Google Translate, as the application videos and some stages were in English. Another unexpected contribution was to gain additional computer skills, such as learning to print out images, learning to take screenshots of designs created by coding, and saving them in a folder by naming these files "e-portfolio." At the beginning of an application, the student demonstrated no proper behaviours, e.g., greetings, sitting, and waiting for lesson, but he did obtain the ability to say, "How are you today?" and wait for the teacher to turn on computer and say "Let's get started." While he did not unmute the video because of his sound sensitivity at the beginning of

an application, he gained the ability to open and listen to the sound at a normal level in the sound-adding activities at the end of the application.

Discussion

The aim of this study was to review the process of teaching coding to a student with ASD and to examine the student's gains in coding on Code.org, along with problems encountered in the process and suggestions for solutions. The findings showed, as a result of the coding instruction process, that the student achieved significant gains. The student successfully completed the first and second levels in the Code.org platform under the researcher's supervision. He completed the last two lessons of his second level without any help and became independent in coding skills. Explicit instruction is preferred for individuals with special needs rather than the constructivist approach applied to peers who typically develop during coding instruction (Basham & Marino, 2013; Israel, Wherfel et al., 2015; Taylor et al., 2017; Taylor, 2018). According to the literature, findings of gradual decrease in the assistance given with explicit instruction and student's increasing independence in our study are consistent with the findings in the literature (Taylor et al., 2017). In addition, in this study, by regulating the help toward the student's needs, he achieved additional gains such as seeing what he could do, developing self-confidence, and learning to ask for help when needed.

Prior to the research, the participant's family thought his interest in technology would lead to academic failure and affect his social interaction as he spent a lot of time on tablets and computers. For this reason, they were restricting his access to technological devices. It was observed that the prospective career planning of the participant increased via transferring his intense attention to coding. The participant made statements such as "I will be an encoder and a software engineer." This supports the view that education within the scope of STEM offers important contributions to individuals' career plans (Israel, Wherfel et al., 2015; Kalelioğlu, 2015; Knight, Wright, Wilson et al., 2019; Lee et al., 2020; Shattuck et al., 2014; Wei et al., 2013). This study also helped the participant with opportunities make a career plan in line with his interests. In addition, his family adopted a positive attitude toward his career plan in this direction and supported his use of technological devices.

It is indicated that the block-based coding platform constructs a basis for advanced computational thinking and programming skills and basic programming logic for students (Israel, Pearson et al., 2015). The studies state that children with typical development learn programming skills with block-based coding applications on platforms such as Code.org, and their motivation to acquire these skills increases (Barradas et al., 2020; Lambić et al., 2020; Munoz et al., 2018). In this process, it was seen that the participant had fun and was highly motivated at all stages, from basic level exercises to the design process. In the research, the student's participation and designs were verbally reinforced. Reinforcement, such as objects and event reinforcers, was not used, as the coding action itself became a natural reinforcement for the student. Knight, Wright, Wilson et al. (2019) suggested that the progress of character with the written code and creating designs were great sources of motivation; thus, no external motivation source was needed.

Knight, Wright, Wilson et al. (2019) and Knight, Wright and Defreese (2019) in their two studies, in which participants had ASD, found that behaviour problems decreased. In this study, although the teacher stated that student's behaviour problems decreased in general education class, it was observed that some behavioural problems occurred in the one-to-one instruction sessions. Problems can be listed as holding the researcher's hand to stop helping, attempting to use both the tablet and desktop computer during the instruction process, refusing to finish the instruction session, opening the computer directly, and entering the coding screen without greeting the teacher. In this case, researchers solved the problems by signing a behaviour contract, removing the tablet computer from the teaching environment, and reducing the teacher's support level.

In addition, several problems occurred on the platform. The biggest limitations included lack of exercises in the researcher's language at some stages, and the lack of language options for the videos prepared for teachers. These limitations were resolved by researchers translating student activities into Turkish. However, this situation may be restrictive for teachers who do not speak English.

There have been some positive student improvements, which were not expected. Researchers and the parents observed the participant's sound sensitivity. Although the student did not allow turning on the volume while watching activity videos at the beginning of the study, it was observed that he listened to sounds that included activities such as adding sound into design. This was one of the contributions of the research. Another contribution was the increase in the student's English vocabulary and the increase in his interest in English. The attempt to name the commands using the Google Translate platform without any help showed that it contributed to the student's problem-solving and independence skills. In the literature (Elsahawy et al., 2020; Geist, 2016; Israel,

Pearson et al., 2015; Israel, Wherfel et al., 2015; Kale & Yuan, 2021; Kalelioğlu, 2015; Taylor et al., 2017; Taylor, 2018), the improvement of problem-solving skills was observed in this case as one of the most important contributions of computational thinking studies.

Another finding obtained in the research was that the coding stages were performed with a tablet computer by the teacher and the desktop computer by the student in the instruction process, which was designed by the explicit instruction method and which increased the student's motivation. This situation gave the student the opportunity to monitor a step twice. The student made coding on the tablet computer by stating that he wanted to use the tablet computer in an instruction session. Here, problems such as clicking on different places on the screen and not seeing all the code lines on the same page caused a decrease in his motivation. As a result, it was determined that, if the student had no limitation in holding a mouse and giving a command with a mouse, it was more appropriate to do this on a desktop computer. Sometimes the use of a mouse can be a difficult skill for students with special needs who are not competent in using computers (Arfe et al., 2020; Israel, Pearson et al., 2015; Wright et al., 2020). It can be said that prerequisite skills are necessary to evaluate students, especially to test these skills before coding instruction. The participant not only gained skills in programming, he also learned new skills, e.g., finding the document he saved on a computer, taking a screenshot, and printing something out. According to the literature, programming teaching not only contributes to career planning in the field of engineering, these skills can also be used in daily life or will be needed in other professions (Israel, Wherfel et al., 2015).

As a result, it is stated that individuals with special needs have limited access to applied programs within the scope of STEM studies. It is also stated that teachers have limitations in terms of transferring abstract subjects to students who have problems in higher-order thinking skills and gaining computer skills while adapting and applying methods in the field of special education (Israel, Wherfel et al., 2015). Although the research on technology and ASD is rich, these studies (for instance, Chia et al., 2018; Pennington, 2010; Wright et al., 2020) are about advanced technologies used to enrich the education provided to individuals with ASD; further, studies in which individuals with ASD are in producer positions remain limited. A review study by Ehsan et al. (2018) on STEM applications for students with ASD demonstrated that the majority of 44 studies included science and mathematics, and only one research took place in the field of technology. In this current study, we found that the student with ASD was successful in block-based coding platform. As Knight, Wright, Wilson et al. (2019) stated, individuals with ASD were successful in block-based coding. There is need for more research on this subject, and it is clearly indicated that this research should not only be on a theoretical basis but also on how to teach computational thinking skills to individuals with special needs in practice, and present it from platforms that include teaching methods.

This study was carried out with a student with ASD who had a great interest in technology. Data obtained should be evaluated within this framework. Studies on teaching students who do not provide prerequisite skills for coding can also be included. In addition, this study was carried out in a one-to-one teaching setting. Individual gains in coding skills instruction include collaboration and coming up with common solutions (Israel, Wherfel et al., 2015; Karp & Maloney, 2013). In this study, these skills were not gained by the participant. Improvements in coding skills in the level of collaboration among students with ASD can be examined in future research.

Authors' Contributions

Researchers worked jointly in process of all parts of the study. The first author contributed more to the analysis and reporting of the study, and the second author contributed more to data collection.

Acknowledgment

Thanks to participant student and his parent for their collaboration with us during the research, to Dr. Onur İşbulan for his valuable contribution and to our undergraduate student İrem Avlar for helping to record videos of instruction stages.

References

- Access Science, Technology, Engineering, and Mathematics. (2007). *AccessSTEM 2007 CBI proceedings*. <https://www.washington.edu/doit/sites/default/files/atoms/files/AccessSTEM%202007%20CBI%20Proceedings.pdf>
- Arfe, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education*, 148, Article 103807. <https://doi.org/10.1016/j.compedu.2020.103807>
- Barradas, R., Lencastre, J. A., Soares, S., & Valente, A. (2020). *Developing computational thinking in early ages: A review of the Code.org platform*. In H. C. Lane, S. Zvacek, & J. Uhomoibhi (Eds.), *Proceedings of the 12th international conference on computer supported education: CSEDU2020* (Vol. 2., pp. 157-168). Science and Technology Publications. http://repositorium.sdum.uminho.pt/bitstream/1822/65432/1/CSEDU_2020_Barradas_Lencastre_Soares_Valente.pdf
- Basham, J. D., & Marino, M. T. (2013). Understanding STEM education and supporting students through universal design for learning. *Teaching Exceptional Children*, 45(4), 8-15. <https://doi.org/10.1177%2F004005991304500401>
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157. <http://dx.doi.org/10.1016/j.compedu.2013.10.020>
- Brantlinger, E., Jimenez, R., Klingner, J., Pugach, M., & Richardson, V. (2005). Qualitative studies in special education. *Exceptional Children*, 71(2), 195-207. <https://doi.org/10.1177%2F001440290507100205>
- Burgstahler, S., Ladner, R. E., & Bellman, S. (2012). Strategies for increasing the participation in computing of students with disabilities. *ACM Inroads*, 3(4), 42-48. <https://doi.org/10.1145/2381083.2381098>
- Chia, G. L. C., Anderson, A., & McLean, L. A. (2018). Use of technology to support self-management in individuals with autism: Systematic review. *Review Journal of Autism and Developmental Disorders*, 5, 142-155. <https://doi.org/10.1007/s40489-018-0129-5>
- Computer Science Teachers Association and International Society of Technology in Education. (2011). *Operational definition of computational thinking for K-12 education*. <https://cdn.iste.org/www-root/Computational Thinking Operational Definition ISTE.pdf>
- Creswell, J. W. (2014). *Educational research: Planning, conducting and evaluating quantitative and qualitative research*. Pearson Education, Inc.
- Ehsan, H., Rispoli, M., Lory, C., & Gregori, E. (2018). A systematic review of STEM instruction with students with autism spectrum disorders. *Review Journal of Autism and Developmental Disorders*, 5, 327-348. <https://doi.org/10.1007/s40489-018-0142-8>
- Eiselt, K., & Carter, P. (2018, October 3-6). *Integrating social skills practice with computer programming for students on the autism spectrum* [Paper presentation]. 48th IEEE Frontiers in Education Conference (FIE), San Jose, United States.
- Elsahawy, M., Aboelnaga, K., & Sharaf, N. (2020, April 27-30). *CodaRoutine: A serious game for introducing sequential programming concepts to children with autism* [Paper presentation]. 11th IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97. <https://doi.org/10.1016/j.compedu.2012.11.016>
- Gay, L. R., Mills, G. E., & Airasan, P. (2006). *Educational research: Competencies for analysis and applications*. Pearson Education Ltd.
- Geist, E. (2016). Robots, programming and coding, oh my! *Childhood Education*, 92(4), 298-304. <https://doi.org/10.1080/00094056.2016.1208008>

- Gribble, J., Hansen, A., Harlow, D., & Franklin, D. (2017, June 27-30). Cracking the code: The impact of computer coding on the interactions of a child with autism. In *Proceedings of the 2017 Conference on Interaction Design and Children* (pp. 445-450). Association for Computing Machinery, Inc. <https://doi.org/10.1145/3078072.3084307>
- Havlaskova, T., Homanova, Z., & Tran, D. (2019, November 21-22). *Developing computational thinking in pre-school children* [Poster presentation]. 17th International Conference on Emerging E-learning Technologies and Applications, Starý Smokovec, Slovakia. <https://ieeexplore.ieee.org/abstract/document/9040113>
- Hwang, J., & Taylor, J. C. (2016). Stemming on stem: A STEM education framework for students with disabilities. *Journal of Science Education for Students with Disabilities*, 19(1), 39-49. <https://doi.org/10.14448/jsesd.06.00017>
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82, 263-279. <https://doi.org/10.1016/j.compedu.2014.11.022>
- Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., & Tapta, T. (2015). Empowering K-12 students with disabilities to learn computational thinking and computer programming. *Teaching Exceptional Children*, 48(1), 45-53. <https://doi.org/10.1177%2F0040059915594790>
- James, W., Lamons, K., Spilka, R., Bustamante, C., Scanlon, E. M., & Chini, J. J. (2019, July 24-25). *Hidden walls: STEM course barriers identified by students with disabilities* [Poster presentation]. Physics Education Research Conference, Provo, Utah, United States. <https://www.compadre.org/per/items/detail.cfm?ID=15285>
- Kafai, Y., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. The MIT Press.
- Kale, U., & Yuan, J. (2021). Still a new kid on the block? Computational thinking as problem solving in Code.org. *Journal of Educational Computing Research*, 59(4), 620-644. <https://doi.org/10.1177/0735633120972050>
- Kalelioglu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>
- Karp, T., & Maloney, P. (2013). Exciting young students in grades K-8 about STEM through an afterschool robotics challenge. *American Journal of Engineering Education*, 4(1), 39-54. <https://doi.org/10.19030/ajee.v4i1.7857>
- Knight, V., Wright, J., & Defreese, A. (2019). Teaching robotics coding to a student with ASD and severe problem behavior. *Journal of Autism and Developmental Disorders*, 49(6), 2632-2636. <https://doi.org/10.1007/s10803-019-03888-3>
- Knight, V., Wright, J., Wilson, K., & Hooper, A. (2019). Teaching digital block-based coding to high school students with ASD and challenging behavior. *Journal of Autism and Developmental Disorders*, 49(8), 3113-3126. <https://doi.org/10.1007/s10803-019-04033-w>
- Kong, S., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, Article 103872. <https://doi.org/10.1016/j.compedu.2020.103872>
- Lambić, D., Đorić, B., & Ivakić, S. (2020). Investigating the effect of the use of Code.org on younger elementary school students' attitudes towards programming. *Behaviour & Information Technology. Advance Online Publication*. <https://doi.org/10.1080/0144929X.2020.1781931>
- Lee, E., Black, M. H., Falkmer, M., Tan, T., Sheehy, L., Bölte, S., & Girdler, S. (2020). "We can see a bright future": Parents' perceptions of the outcomes of participating in a strengths-based program for adolescents with autism spectrum disorder. *Journal of Autism and Developmental Disorders*, 50(9), 3179-3194. <https://doi.org/10.1007/s10803-020-04411-9>
- Lindsay, S., & Hounsell, K. G. (2017). Adapting a robotics program to enhance participation and interest in STEM among children with disabilities: A pilot study. *Disability and Rehabilitation: Assistive Technology*, 12(7), 694-704. <https://doi.org/10.1080/17483107.2016.1229047>

- Ludi, S., & Reichlmayr, T. (2011). The use of robotics to promote computing to pre-college students with visual impairments. *Acm Transactions on Computing Education*, 11(3), 1-20. <https://doi.org/10.1145/2037276.2037284>
- Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, 19(4), 790-824. <https://citejournal.org/volume-19/issue-4-19/general/preparing-elementary-school-teachers-to-teach-computing-coding-and-computational-thinking>
- Meccawy, M. (2017). Raising a programmer: Teaching Saudi children how to code. *International Journal of Educational Technology*, 4(2), 56-65. <https://educationaltechnology.net/ijet/index.php/ijet/article/view/25>
- Miles, M. B., Huberman, A. M., & Saldana, J. (2014). *Qualitative data analysis: A methods sourcebook* (3rd ed.). Sage Publications, Inc.
- Munoz, R., Villarroel, R., Barcelos, T. S., Riquelme, F., Quezada, Á., & Bustos-Valenzuela, P. (2018). Developing computational thinking skills in adolescents with autism spectrum disorder through digital game programming. *IEEE Access*, 6, 63880-63889. <https://doi.org/10.1109/access.2018.2877417>
- National Education Association. (2012). *Preparing 21st century students for a global society: An educator's guide to the "four cs"*. https://www.aledoisd.org/cms/lib/TX02205721/Centricity/Domain/2020/Preparing21C_Learners.pdf
- National Science Foundation. (2009). *A week to focus on computer science education*. http://www.nsf.gov/news/news_summ.jsp?cntn_id=116059
- Newman, L., Wagner, M., Knokey, A. M., Marder, C., Nagle, K., Shaver, D., Wei, X., Cameto, R., Contreras, E., Ferguson, K., Greene, S., & Schwarting, M. (2011). *The post-high school outcomes of young adults with disabilities up to 8 years after high school: A report from the national longitudinal transition study-2*. https://nlts2.sri.com/reports/2011_09_02/nlts2_report_2011_09_02_complete.pdf
- Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. *International Journal of Mobile Learning and Organisation*, 10(3), 187-202. <https://doi.org/10.1504/IJMLO.2016.077867>
- Patton, M. Q. (2002). *Qualitative research and evaluation methods* (3rd ed.). Sage Publications, Inc.
- Pennington, R. C. (2010). Computer-assisted instruction for teaching academic skills to students with autism spectrum disorders: A review of literature. *Focus on Autism and Other Developmental Disabilities*, 25(4), 239-248. <https://doi.org/10.1177%2F1088357610378291>
- Ratcliff, C. C., & Anderson, S. E. (2011). Reviving the turtle: Exploring the use of logo with students with mild disabilities. *Computers in the Schools*, 28(3), 241-255. <https://doi.org/10.1080/07380569.2011.594987>
- Samson, F., Mottron, L., Souliere, I., & Zeffiro, T. A. (2012). Enhanced visual functioning in autism: An ALE meta-analysis. *Human Brain Mapping*, 33(7), 1553-1581. <https://doi.org/10.1002/hbm.21307>
- Shattuck, P. T., Steinberg, J., Yu, J., Wei, X., Cooper, B. P., Newman, L., & Roux, A. M. (2014). Disability identification and self-efficacy among college students on the autism spectrum. *Autism Research and Treatment*, Article 924182. <https://doi.org/10.1155/2014/924182>
- Sullivan, A. A., Bers, M. U., & Mihm, C. (2017, July 13-15). *Imagining, playing, and coding with KIBO: Using robotics to foster computational thinking in young children* [Paper presentation]. Conference Proceedings of International Conference on Computational Thinking Education, Hong Kong, China.
- Taylor, M. S. (2018). Computer programming with pre-K through first-grade students with intellectual disabilities. *The Journal of Special Education*, 52(2), 78-88. <https://doi.org/10.1177%2F0022466918761120>
- Taylor, M. S., Vasquez, E., & Donehower, C. (2017). Computer programming with early elementary students with Down Syndrome. *Journal of Special Education Technology*, 32(3), 149-159. <https://doi.org/10.1177%2F0162643417704439>
- Tekin, E., & Kırcaali-İftar, G. (2001). *Özel eğitimde yanlışsız öğretim yöntemleri [Errorless teaching methods in special education]*. Nobel Yayın Dağıtım.

- Thomas, T. A. (2014). *Elementary teachers' receptivity to integrated science, technology, engineering, and mathematics (STEM) education in the elementary grades* [Doctoral dissertation, University of Nevada]. <http://hdl.handle.net/11714/2852>
- Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014, October 22-25). *A review of games designed to improve introductory computer programming competencies* [Paper presentation]. Frontiers in Education Conference, Madrid, Spain.
- Vilorio, D. (2014). STEM 101: Intro to tomorrow's jobs. *Occupational Outlook Quarterly*, 58(1), 3-12. https://www.citadel.edu/root/images/career_center/spring2014ooq.pdf
- Wei, X., Yu, J. W., Shattuck, P., McCracken, M., & Blackorby, J. (2013). Science, technology, engineering, and mathematics (STEM) participation among college students with an autism spectrum disorder. *Journal of Autism and Developmental Disorders*, 43(7), 1539-1546. <https://doi.org/10.1007/s10803-012-1700-z>
- Wright, J. C., Knight, V. F., & Barton, E. E. (2020). A review of video modelling to teach STEM to students with autism and intellectual disability. *Research in Autism Spectrum Disorders*, 70, Article 101476. <https://doi.org/10.1016/j.rasd.2019.101476>



Otizm Spektrum Bozukluğu Olan Bir Öğrenciye Blok Tabanlı Kodlama Öğretim Süreci

Canan Sola-Özgüç¹

Damla Altın²

Öz

Giriş: Bu araştırmanın amacı, otizm spektrum bozukluğu (OSB) olan bir öğrenciye kodlama öğretimi sürecinin ayrıntılı bir şekilde aktararak, öğrencinin kodlamada elde ettiği kazanımların, süreçte karşılaşılan problemlerin ve çözüm önerilerinin incelenmesidir.

Yöntem: Araştırma durum araştırması olarak desenlenmiştir ve üç aylık bir öğretim sürecini kapsamıştır. Bu süreçte 10 yaşındaki otizm spektrum bozukluğu olan bir erkek öğrenciye kodlama becerilerinin doğrudan öğretim yöntemiyle hazırlanan öğretim oturumları, bu çalışmanın katılımcıya katkıları elde edilen çeşitli verilerle analiz edilmiştir. Veriler, içerik analizi ile Nvivo 10 programı kullanılarak analiz edilmiştir.

Bulgular: Bulgular, kodlama öğretimi sürecinin sonucunda katılımcıda önemli kazanımlar elde edildiğini göstermiştir. Elde edilen kazanımlarla katılımcının motivasyonunun arttığı görülmüştür. Katılımcı ileride kodlamayı meslek olarak yapmak istediğini belirtmiştir.

Tartışma: Bu çalışmada katılımcının programlama becerileri kazanmanın yanında bilgisayara kaydettiği belgeyi bulma, ekran görüntüsü alma ve bir şeyler yazdırma gibi yeni beceriler öğrendiği belirlenmiştir. Bu konuda yapılan çalışmalar da programlama öğretiminin mühendislik alanında kariyer planlamasına katkı sağlamasının yanında edinilen becerilerin günlük hayatta da kullanılabilirliğini veya diğer mesleklerde de ihtiyaç duyulacağını ifade etmektedir. Bu çalışmada da OSB olan bireylerin teknolojiyi sadece tüketici konumunda kullanmaları değil, bilgi işlemsel düşüncelerini geliştirerek teknoloji alanında üretken bireyler olmaları gerektiği vurgulanmıştır.

Anahtar kelimeler: STEM, otizm spektrum bozukluğu, kodlama, code.org, bilgi işlemsel düşünme, durum araştırması.

Atf için: Sola-Özgüç, C., & Altın, D. (2022). Otizm spektrum bozukluğu olan bir öğrenciye blok tabanlı kodlama öğretim süreci. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Özel Eğitim Dergisi*, 23(3), 565-594. <https://doi.org/10.21565/ozelegitimdergisi.822554>

¹**Sorumlu Yazar:** Dr. Öğr. Üyesi, Sakarya Üniversitesi, E-posta: csola@sakarya.edu.tr, <https://orcid.org/0000-0002-4716-9028>

²Arş. Gör., Sakarya Üniversitesi, E-posta: altin@sakarya.edu.tr, <https://orcid.org/0000-0001-8091-7625>

Giriş

Çağımızda dijitalleşmenin artmasıyla birlikte, günümüz öğrencileri teknolojiye daha çabuk uyum sağlamaktadırlar. Bu açıdan düşünüldüğünde bu jenerasyonun bilgisayar becerilerinin edinmesinin yanında teknoloji alanında üretici olmaları konusunda eğilimlerin olduğu görülmektedir. Bir başka ifade ile günümüzde insanlardan yalnızca tüketici değil aynı zamanda üretici olmaları beklenmektedir (Ehsan vd., 2017; Kalelioğlu, 2015). Bu amaçla eğitim programlarında İngilizce Science (bilim), technology (teknoloji), engineering (mühendislik) ve mathematics (matematik) kelimelerinin baş harflerinden oluşturulan STEM uygulamaları ile okul öncesinden yüksek öğretime kadar her yaşta bireyin bu alanlara hazırlanırken, bir taraftan da bu alanların bakış açısı ile günlük yaşam problemlerini çözme becerisini kazanmaları hedeflenmektedir. (Hwang & Taylor, 2016; Thomas, 2014). Öğrencilerin çağa ayak uydurabilmeleri ve iyi bir performans sergileyebilmeleri için STEM eğitime ihtiyaçları vardır. Son 20 yılda STEM'e dayalı eğitim programlarının uygulamadaki artışının yanında STEM'in sağladığı istihdam olanaklarının da arttığı gözlenmektedir. Böylece STEM'e dayalı iş olanakları ve STEM eğitim programlarına ilginin arttığı görülmektedir (Ehsan vd., 2017). STEM uygulamalarının özel gereksinimli bireylerin eğitimlerinde yer verilmesi gerekliliği, özel gereksinimli bireylere yapılan STEM uygulamaları, STEM'in özel gereksinimli bireylere uygulanması sürecinde karşılaşılan sınırlılıklar çalışma konuları arasında yer almaktadır (James vd., 2019).

STEM uygulamalarında her öğrenciye eşit öğrenme fırsatı sunulması hedeflense de bu uygulamalarda özel gereksinimli öğrenciler, tipik gelişim gösteren çocuklara kıyasla daha düşük performans göstermektedirler. Bu durum özel gereksinimli bireylerin hem akademik hem de kariyer gelişimlerini olumsuz yönde etkilemektedir (Burgstahler vd., 2012). Bu sebeple ilerideki kariyer planlarına katkı sağlaması ve yalnızca tüketen değil, aynı zamanda üreten bireyler olabilmeleri için, özel gereksinimli bireylerin STEM uygulamalarına katılmaları önem taşımaktadır (AccessStem, 2007). Amerikan Ulusal Özel Eğitim Araştırmaları Birimi (National Center for Special Education Research [NCSE]) tarafından 2011 yılında yayınlanan rapora göre istihdam edilen özel gereksinimli bireylerin %4'ten daha azının STEM alanında çalıştıkları görülmektedir. STEM alanındaki işlerde 2022 yılına kadar 5.5 milyon insan gücüne ihtiyaç duyulacağı öngörülmektedir (Newman vd., 2011; Vilorio, 2014). Geleceğin bu şekilde öngörüldüğü bir ortamda özel gereksinimli bireylerin bu alanlarda istihdam edilebilmeleri için STEM eğitime tipik gelişim gösteren bireyler kadar erişebilmesi önem taşımaktadır. STEM eğitimiyle eğitimcilere yeni standartlar sunularak özel gereksinimli bireylerin de STEM uygulamalarına erişimi vurgulanmaktadır. Bilgi işlemsel düşünme becerileri kapsamında öğretilen becerilerin bireylerin kariyer edinmesine katkı sağlamasının yanında yaşamındaki farklı alanlara da bu becerileri transfer edebilmelerini sağlayacağı düşünülmektedir. Amerikan Ulusal Bilim Birliği (The National Science Foundation) (2009), STEM kapsamında edinilen becerilerin bilgisayar ve programlama iş alanlarında kullanımının yanında, edinecekleri diğer meslekler için de önemli katkıları olacağını belirtmektedir. Zorunlu eğitim programında verilen programlama becerisinin, (1) problem çözme, iş birliği içerisinde araştırma yapma becerilerini geliştirdiğini, (2) üst düzey düşünme becerileri konusunda temel oluşturduğunu, (3) iş birliği içinde problem çözmeyi geliştirdiğini ve (4) bilgisayar bilimi hakkında olumlu tutumlar geliştirdiğini, (5) öğrencileri motive ettiğini göstermiştir (Fessakis vd., 2013; Havlaskova vd., 2019; Kafai & Burke, 2014; Kalelioğlu, 2015). Bu katkılar göz önünde bulundurulduğunda programlama ile ilgili deneyimlerin tipik gelişen ya da özel gereksinimli öğrencileri için hem kariyer yolu açmış hem de geniş bir eğitsel yarar sağlamış olacaktır.

Yirmi birinci yüzyıl becerileri olarak ifade edilen bilgi işlemsel düşünme becerilerinin öğretim programlarına entegre edilerek kazandırılması gerekir (Kong vd., 2020). Bilimsel çalışmalar bu konudaki önemi vurgulasa da öğretmenlerin bilgi işlemsel düşünmenin ne anlama geldiği konusunda bilgilerinin sınırlı olduğunu göstermektedir (Israel, Wherfel vd., 2015; Kong vd., 2020; Mason & Rich, 2019). Bilgi işlemsel düşünme, Amerikan Bilgisayar Öğretmenleri Birliği ve Uluslararası Eğitimde Teknoloji Derneği (International Society for Technology in Education [ISTE], 2020, s. 1) tarafından şu şekilde tanımlanmaktadır:

“Problem çözme becerisi, karşılaştığımız problemler karşısında verileri belirli bir mantık içinde organize ederek analiz etmek; model ve simülasyonlarla veriyi soyutlaştırmak; algoritmik düşünme ile çözümleri otomatikleştirmek; kaynakların etkili ve verimli şekilde kullanılabilmesi için olası çözümleri uygulamak ve çözüm sürecini daha geniş çaplı problemlerin çözümüne genellemek ve aktarmak için bilgisayar ve diğer araçlardan yararlanmaktır.”

Bilgi işlemsel düşünmenin tanımı bağlamında özel gereksinimli öğrencilerin karmaşık problem çözme ve soyut akıl yürütmede zorlanacakları düşünülmektedir. Bu nedenle de özel gereksinimli öğrenciler ile çalışan öğretmenlerin bu durumda öğrencilerini nasıl destekleyeceklerini dikkate almaları gerekmektedir. Böylece özel

gereksinimli olan öğrencilerin de programlama ile ilgili eğitimlerden faydalanmalarını sağlamış olacaktırlar (Israel, Pearson vd., 2015; Taylor, 2018).

Programlama bir diğer ismiyle kodlama; problemleri çözmek, insan ile bilgisayar etkileşimini sağlamak ve belirli bir görevi bilgisayarlar tarafından gerçekleştirmek için çeşitli komut setleri ile yapılan uygulama ve geliştirme sürecidir. Günümüzde kodlama, her öğrencinin kazanması gereken temel bir beceri olarak görülmektedir (Kafai & Burke, 2014). Kodlama öğretimi becerilerinin erken yaşlardan itibaren Scratch, Code.org, Kodable, The Foes, Tynker gibi farklı platformlar aracılığı ile tipik gelişen öğrencilere öğretildiği ve bu öğretim sonrasında öğrencilerde bilgi işlemsel düşünme becerilerinde gelişmeler olduğu görülmektedir (Bers vd., 2013; Geist, 2016; Papadakis vd., 2016; Ratcliff & Anderson, 2011; Sullivan vd., 2017). Özel gereksinimli öğrenciler söz konusu olduğunda bu alanda çalışmaların sınırlı olduğu görülmektedir. Yapılan sınırlı araştırmalar bilişsel alanda sınırlılık yaşayan zihin yetersizliği olan öğrencilerin ve OSB olan öğrencilerin de bu becerileri edinebildiğini göstermektedir (Israel, Pearson vd., 2015; Knight, Wright & Defreese, 2019; Knight, Wright, Wilson vd., 2019; Ratcliff & Anderson, 2011; Taylor vd., 2017; Taylor, 2018). Bu çalışmalarda özel gereksinimli bireylerin de üretimde yer alması gerektiği belirtilmekte ve daha fazla çalışmaya ihtiyaç olduğu vurgulanmaktadır. Ayrıca OSB’li bireylerin görsel işlemlerde güçlü olmaları, detaylara odaklanma, bir konuda sürekliliği sağlama konusunda becerilerinin oldukça iyi olması (Samson vd., 2012) nedeni ile OSB’li öğrencilerin eğitiminde ve istihdamında kodlamanın önemli bir yer tutacağı söylenebilir. Diğer yetersizlik alanları ile kıyaslandığında bilgisayar bilimlerini % 16.22 ile en fazla OSB’li bireyler tercih etmektedirler. Gelecekte daha fazla OSB’li bireyin bilgisayar bilimlerini tercih edeceği öngörülmektedir (Shattuck vd., 2014; Wei vd., 2013). Bu potansiyelin farkında olan büyük teknoloji şirketleri de OSB’li bireylerin küçük detayları fark edebilme, kalıpların dışına çıkma, titiz çalışma gibi özelliklerinden faydalanma amacıyla uygun istihdam programları geliştirmekte, OSB’li bireyleri çalışma ortamlarında desteklemektedirler (Eiselt & Carter, 2018; Knight, Wright, Wilson vd., 2019). OSB’li bireylerin bu ilgileri göz önünde bulundurulduğunda, erken yaşlarda bilgisayar bilimlerine yönelik eğitim almalarının gelecekte bilgisayar bilimleri kapsamında bir meslek alanına yönlendirilebilmeleri açısından önem taşımaktadır (Knight, Wright & Defreese, 2019).

Öğretmenlerin, bilgi işlemsel düşünmenin bilişsel açıdan sınırlılık yaşayan öğrencilere öğretimi konusunda bilgi ihtiyaçlarının olduğu belirtilmekte, daha fazla araştırma yapılması gerekliliğini ortaya koymaktadır (Taylor, 2018). Bu çalışmada hafif düzeyde zihin yetersizliğinin eşlik ettiği OSB olan bir öğrenciye blok tabanlı kodlama sürecinin nasıl düzenlendiği, ne tür sorunlarla karşılaşıldığı ve karşılaşılan bu sorunların nasıl çözümlendiğine ilişkin bir durum çalışmasının yapılması hedeflenmiştir. Bu temel hedef doğrultusunda aşağıdaki alt soruların yanıtlanması hedeflenmiştir.

1. OSB olan bir öğrenciye blok tabanlı kodlama öğretimi nasıl gerçekleştirmiştir?
2. OSB olan bir öğrencinin blok tabanlı kodlama öğretiminde başarısı nedir?
3. OSB olan bir öğrenciye blok tabanlı kodlama öğretim sürecinde karşılaşılan problemler nelerdir? Nasıl çözümlenmiştir?
4. Blok kodlama öğretiminin OSB olan öğrenciye ne gibi katkıları olmuştur?

Yöntem

Bu araştırmanın amacı hafif düzeyde zihin yetersizliğinin eşlik ettiği OSB olan bir öğrenciye Code.org platformu üzerinde birinci ve ikinci kurs düzeyinde kodlama öğretimi sürecinin nasıl gerçekleştirildiğini aktarmaktır. Çalışma, OSB olan bir öğrenci ile sınırlı olup, kodlama öğrenme sürecinin detaylı incelenmesi amacıyla durum çalışması yöntemi seçilmiştir.

Katılımcı

Araştırmanın katılımcısı 10 yaşında hafif düzeyde zihin yetersizliğine sahip OSB olan bir erkek öğrencidir. Öğrenci, bir devlet ilkokulunun dördüncü sınıfında tam zamanlı kaynaştırma eğitimi almaktadır. Ayrıca haftada iki saat devlet tarafından desteklenen bir eğitim kurumundan akademik beceriler konusunda destek eğitimi almaktadır. Öğrencinin Wechsler Intelligence Scale for Children IV (WISC-R IV) testinden elde edilen genel zeka puanı 74’tür. Testin alt boyutlarından sözel-dilsel alanda 63, performans boyutunda 91 puan almıştır. Tıbbi değerlendirme raporunda “otizm spektrum bozukluğu” tanısı bulunmaktadır.

Katılımcının yoğun ilgi duyduğu alan teknolojik araçlardır. Teknolojiye olan ilgisi yanında; akademik davranışlarında gerileyeceği düşüncesi ile teknolojik aletlere ulaşımı ailesi tarafından sınırlandırılmıştır. Öğrencinin, teknolojik araç gördüğü anda kimin olduğu fark etmeksizin, izin almadan o aracı inceleme, kişisel

alan ihlali, ders sırasında sınıfta gezinme, yüksek seste öfke nöbeti geçirme gibi davranış problemleri görülmektedir.

Katılımcının ilkokul öğretmeni öğrencinin ilgisine yönelik etkinlik geliştiremediği için yazarlardan yardım talep etmiştir. Katılımcıya ilgisi doğrultusunda bilgi işlemsel bir uygulama yapılması hedeflenmiştir. Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde görev yapan bir öğretim üyesine danışılmıştır. Öğretim üyesi öğrencinin özellikleri doğrultusunda Code.org’u önermiştir.

Ortam ve Materyaller

Araştırma bir devlet üniversitesinin özel eğitim biriminde gerçekleştirilmiştir. Bu araştırma süresince düşük düzey ve ileri düzey teknolojik araçlar kullanılmıştır. Düşük düzey teknolojik araçlar, beyaz tahta, tahta kalemi, kâğıt, bilgisayarsız etkinlikler için gerekli olan kürdan, marşmelov, iskambil kartı şeklindedir. İleri düzey teknolojik araçlar olarak Samsung tablet, HP masa üstü bilgisayarı ve HP yazıcı kullanılmıştır.

Kullanılan Platform: Code.org

Code.org, her öğrencinin bilgisayar bilimini öğrenmesini amaçlayan, kar amacı gütmeyen, 2013 yılından bu yana hizmet veren bir platformdur. Code.org *bilgisayar bilimi temelleri, bağımsız dersler ve kod saati* kurslarından oluşmaktadır. Code.org üzerindeki etkinlikler 4-18+ yaş aralığındaki kullanıcılara yönelik olarak hazırlanmıştır.

Verilerin Toplanması

Araştırmanın verilerini: (1) gözlem, (2) aile, öğrencinin öğretmeni ve öğrencinin kendisi ile yapılan yarı-yapılandırılmış görüşmeler, (3) öğrencinin öğretmeni ve ailesiyle gerçekleştirilen yapılandırılmamış görüşmeler, (4) kontrol listeleri, (5) araştırmacıların planlama ve değerlendirme toplantı tutanakları, (6) öğrencinin ürünleri ve (7) araştırmacı günlüğü oluşturmaktadır. Sıralanan bu veri kaynaklarından elde edilen veriler ve veri kaynağı türü Tablo 1’de belirtilmiştir.

Tablo 1

Veriler ve Veri Kaynağı Türü

Amaç	Veri toplama yöntemi	Veri kaynağı
Öğrencinin ön koşul becerilerini değerlendirme	Gözlem Kontrol listesi	Görüntü kaydı
Öğrenciye ilişkin detaylı bilgilerin alınması	Babayla yapılan yarı-yapılandırılmış görüşme	Ses kaydı
Öğrenciye ilişkin detaylı bilgilerin alınması	Öğretmenle yapılan yarı-yapılandırılmış görüşme	Ses kaydı
Öğretim süreci	Gözlem Araştırmacı günlüğü Değerlendirme toplantıları Öğrenci ürünleri	Görüntü kaydı Metin belgesi Ses kaydı Portfolyo

Araştırmacılar

Bu araştırma iki araştırmacı tarafından yürütülmüştür. Birinci yazarın rolü; öğretim planlamasını yapma (bir oturumda kaç aşamanın yapılacağı, hangi aşamalarda hangi ipuçlarının sunulacağını planlanması), uygulamayı gözleme, uygulamanın görüntü kaydını alma, günlük yazmadır. İkinci yazarın rolü, öğretim planlamasını yapma (bir oturumda kaç aşamanın yapılacağı, hangi aşamalarda hangi ipuçlarının sunulacağını planlanması) uygulama için gerekli materyalleri planlama ve temin etme, uygulamayı gerçekleştirme, günlük yazmadır.

Araştırma Süreci

Araştırma süreci uygulama öncesi öğrenciye ilişkin detaylı verilerin elde edilmesi ve kodlama eğitiminin verildiği uygulama süreci olarak iki aşamada gerçekleştirilmiştir. Birinci aşamada öğrencinin kodlama eğitimi için ön koşul becerileri değerlendirilmiş, aile ve öğretmeninden veriler elde edilerek uygulama süreci planlanmıştır. Öğretim ortamı (Code.org), öğretim yöntemi, kullanılacak araçlar (masaüstü bilgisayar, tablet bilgisayar, vb.) planlandıktan sonra ikinci aşamada uygulama gerçekleştirilmiş, öğrenci performansı ve karşılaşılan durumlara ilişkin kararlar alınarak uygulamaya dair veriler toplanmıştır. Araştırma süreci OSB olan bir öğrenciye blok tabanlı kodlama öğretimi nasıl gerçekleştirilmiştir sorusunu yanıtladığı için bulgular başlığı altında detaylı bir şekilde aktarılmıştır.

Verilerin Analizi

Verilerin analizi hem uygulama sürecinde hem de uygulama sonrasında gerçekleştirilmiştir. Uygulama sürecinde araştırmacılar, günlük olarak yaptıkları uygulamayı irdeleyerek, veriler bağlamında araştırmayı yönlendirmişlerdir. Uygulama sonrasında yapılan analiz süreci ise verilerin analize hazırlanması ve analiz edilmesi aşamalarından oluşmuştur. Verilerin analize hazırlanma sürecinde öğretmen ve aile görüşmelerinde elde edilen ses kayıtlarının dökümleri yapılmıştır. Yapılan dökümler daha önce iki farklı araştırma sürecinde aynı görevi gerçekleştirmiş bir lisans öğrencisi tarafından kontrol edilmiştir.

Araştırmacı günlüğü, aile görüşmesi ve öğretmen görüşmesi dökümleri her iki yazar tarafından okunmuş ve kodlanmıştır. Her iki yazarın araştırma sorularına cevap oluşturacağını düşündüğü tarihlerdeki görüntü kayıtları izlenerek detaylı analizleri yapılmıştır. Ses kayıtlarının dökümleri, araştırmacı günlüklerinin metinleri NVivo 10 programına aktarılmış, bu veriler NVivo 10 programı ile analiz edilmiştir. Bu analizden elde edilen temalar araştırma sorularına cevap olacak şekilde raporlaştırılmıştır.

Geçerlik, Güvenirlik ve Etik

Araştırmada, nitel araştırmaların geçerliliği için dikkat edilmesi gereken bazı durumlar göz önünde bulundurulmuştur. Bunun için;

1. Yarı-yapılandırılmış görüşme sorularının hazırlanmasında, özel eğitim alanında ve nitel araştırma yöntemi konusunda çalışmaları olan üç uzmandan uzman görüşü alınmıştır.
2. Öğrencinin bilgisayar kullanımı kodlama konusundaki var olan performans düzeyinin belirlenmesi için üç alan uzmanından uzman görüşü alınmıştır.
3. Çeşitli veri kaynakları kullanılarak veri çeşitlemesi sağlanmıştır (Brantlinger vd., 2005; Creswell, 2014; Gay vd., 2006; Miles vd., 2014).
4. Araştırmanın güvenilirliği için, bağımsız olarak iki araştırmacı verileri kodlayarak kategorileri oluşturmuşlardır. Kategorileri karşılaştırarak, ortak ve ortak olmayan kategorileri irdelemişlerdir. Yapılan bu niteliksel tutarlılık çalışmasının yanında niceliksel güvenirlilik gerçekleştirilmiştir. Araştırmanın güvenilirliği Miles ve Huberman (1994) tarafından önerilen Güvenirlilik = Görüş Birliği / (Görüş Birliği + Görüş Ayrılığı) x 100 formülü ile hesaplanmıştır (akt., Patton, 2002). İkinci yazar ile tema-alt temalar bağlamında yapılan güvenirlilik %98 olarak hesaplanmıştır.
5. Araştırmacının uygulamayı öğretim planına uygun olarak yürütüp yürütmediğini belirlemek ve plana ne derece bağlı kaldığını belirlemek amacıyla Uygulama Güvenirliliği Formu kullanılarak uygulama güvenirliliği verisi alınmıştır. Hesaplamalar doğrultusunda uygulama güvenirliliği %94 olarak belirlenmiştir.
6. Araştırma, 07/03/2020 tarihli 61923333/050.99/sayılı Sakarya Üniversitesi Etik Kurul Onayı ile gerçekleştirilmiştir.

Bulgular

OSB Olan Bir Öğrenciye Blok Tabanlı Kodlama Öğretimi Nasıl Gerçekleştirmiştir?

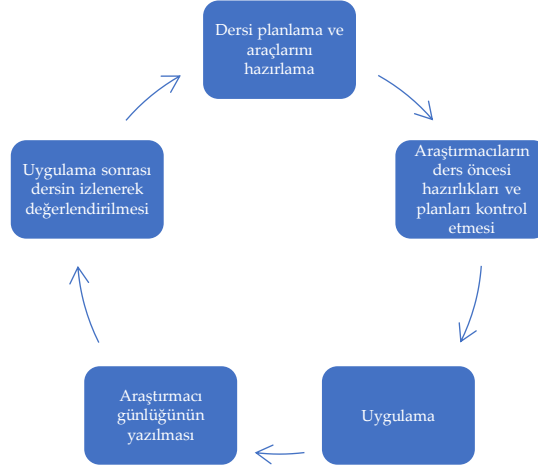
Araştırmanın uygulama sürecinden önce öğrencinin uygulama için (Code.org platformunda öğretim yapılabilmesi için) gerekli ön koşul becerileri sağlayıp sağlamadığına dair araştırmacılar tarafından geliştirilen kontrol listesi uygulanmıştır. Kontrol listesi, 97 madde, dokuz bölümden (renkler, varlıklar arası ilişkiler, sayılar, işlem bilgisi, geometri, yön kavramları, fare ve klavye kullanımı ve internet kullanımı) oluşmaktadır. Kodlama öğretimine başlamadan önce gerekli olan bu becerilerden ne kadarını gerçekleştirdiğini belirlemek üzere değerlendirme oturumları gerçekleştirilmiştir (19.02.2019 ve 21.02. 2019 tarihli görüntü kayıtları). Performans değerlendirmesi sonrasında öğrencinin gerekli ön koşul becerilerden dairenin çevresini hesaplama, 60 derecelik açı çizme becerisine sahip olmadığı, diğer becerileri yerine getirebildiği belirlenmiştir. Gerçekleştiremediği becerilerin gerekli olduğu kodlama aşamasında öğretiminin yapılmasına yansıtma toplantısında karar verilmiştir (21.02.2019 tarihli yansıtma toplantısı). Performans belirleme oturumu sonrasında öğrenci ile yapılan deneme dersinde öğrencinin Kurs 2'yi yapabileceği ancak Kurs 1 itibarı ile başlarsa programlama mantığını daha iyi anlayacağı görüşü ile araştırmacılar öğretime Kurs 1'den başlamışlardır.

Bu araştırmada kodlama öğretimi Code.org üzerinde ikinci araştırmacı tarafından birebir öğretim şeklinde gerçekleştirilmiştir. Araştırmacılar öğrenciye öğretim yöntemi olarak, doğrudan öğretim yöntemini

benimsemişlerdir. Bu doğrultuda Code.org'un Kurs 1 sayfasında yer alan aşamaları model olma, rehberli uygulamalar ve bağımsız uygulamalar olacak şekilde bölümlere ayırarak uygulama öncesinde planlamasını yapmışlardır. Haftada iki farklı gün birer saat olarak yapılan uygulamanın süreçlerine Şekil 1' de yer verilmiştir.

Şekil 1

Uygulama Süreci



Birinci derste araştırmacı tablet bilgisayarda model olmuş, masa üstü bilgisayarda öğrencinin aynı basamağı uygulamasını istemiştir. Birinci ders sonunda öğrencinin model olma basamağına ihtiyaç duymadığı görüntü kayıtları izlenerek karar verilmiştir. Bundan sonraki derslerde model olma aşama sayısı azaltılarak ağırlıklı olarak rehberli uygulama ve bağımsız uygulama aşamalarına yer verilmiştir.

Kurs 1, yedi oturumda; Kurs 2, 11 oturumda gerçekleştirilmiştir. Oturumlarda yer alan ders sayısı ortalama ders süresinin 40-50 dakika süreceği şekilde planlanmıştır. Buna göre bazı oturumlarda tek ders işlenmiştir. Örneğin Kurs 2, "11. Sanatçı: Hata Ayıklama" dersi 25.04.2019 tarihinde tek ders olarak işlenmiştir. Öte yandan 06.05.2019 tarihli Kurs 2, "14. İkili Bileklikler"; "15. Büyük Olay" ve "16. Flappy" derslerinin tümü 40 dakika sürmüş ve üç ders olarak işlenmiştir. Derslerin planlanmasında süre ve ders konu bütünlüğünün olması dikkate alınmıştır. Kurs 1 ve Kurs 2'ye ilişkin detaylar Tablo 2'de verilmiştir.

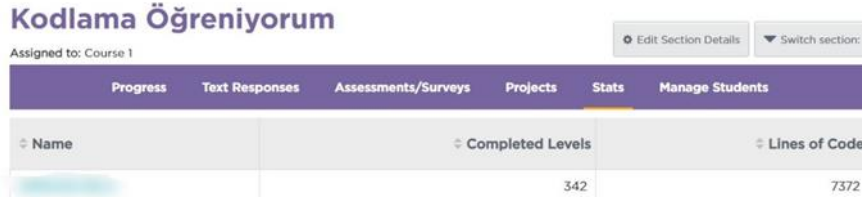
Tablo 2

Code.org Platformunda İşlenen Dersler ve Süreleri

Tarih	Ders adı	Toplam süre
01.03.2019	Kurs 1: Ders 1, 2, 3, 4	00:31:18
04.03.2019	Kurs 1: Ders 5, 6, 7	00:46:01
07.03.2019	Kurs 1: Ders 8, 9, 10	00:44:38
11.03.2019	Kurs 1: Ders 12, 13	00:27:02
14.03.2019	Kurs 1: Ders 14	00:42:08
18.03.2019	Kurs 1: Ders 15, 16	00:48:24
21.03.2019	Kurs 1: Ders 17, 18	00:45:13
25.03.2019	Kurs 2: Ders 1, 2, 3	01:11:24
28.03.2019	Kurs 2: Ders 4	00:33:09
04.04.2019	Kurs 2: Ders 5, 6	00:48:53
11.04.2019	Kurs 2: Ders 7	00:10:37
15.04.2019	Kurs 2: Ders 8	00:29:00
18.04.2019	Kurs 2: Ders 9, 10	00:25:45
25.04.2019	Kurs 2: Ders 11	00:38:19
02.05.2019	Kurs 2: Ders 12, 13	00:48:33
06.05.2019	Kurs 2: Ders 14, 15, 16	01:38:51
09.05.2019	Kurs 2: Ders 17	01:18:24
13.05.2019	Kurs 2: Ders 18, 19	01:21:55

OSB Olan Bir Öğrencinin Blok Tabanlı Kodlama Öğretiminde Başarısı Nedir?

Bu uygulama sonucunda öğrenci 342 seviye tamamlamış, 7372 kod satırı kullanmıştır. Şekil 2'de platformdaki veriler sunulmuştur.

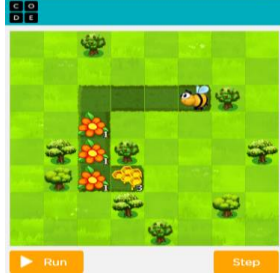
Şekil 2*Katılımcının Uygulama Sürecindeki Performansı*

Katılımcı Kurs 1 ve Kurs 2 olmak üzere toplamda iki kurs ve 37 ders bitirmiştir. Kurs 1'deki 11. ders "Spelling Bee" adında İngilizce Puzzle etkinliği olması nedeniyle gerçekleştirilememiştir. Verilerde bu aşamanın yapılmadığı görülmektedir. Kurs 1'de iki aşamadan oluşan 17. dersin bir kısmının yapılmamış olarak görünmesinin nedeni, bu dersin bir aşamasının bilgisayarsız etkinlik olmasıdır. Şekil 3'te katılımcının Kurs 1'de tamamladığı, Şekil 4'te ise katılımcının Kurs 2'de tamamladığı dersler görülmektedir.

Şekil 3*Katılımcının Kurs 1'de Tamamladığı Dersler***Şekil 4***Katılımcının Kurs 2'de Tamamladığı Dersler*

Araştırmacılar önceki sayfalarda belirtildiği üzere öğretime, öğrencinin yaş aralığında olan seviyeden değil başlangıç seviyesinden başlamışlardır. Uygulama sırasında öğrencinin Kurs 1'de sıkılmadığı ve aşamalarda çok az da olsa hatalar yapıp hatalarını bulduğu gözlenmiştir. Ayrıca öğrencinin "Ben Kurs 1'i de sevdim" şeklindeki ifadesi, bu aşamadan başlamanın öğrenci için sıkıcı değil daha çok yapıcı olduğu kanısını desteklemektedir. Bu bağlamda daha önce kodlama konusunda ön çalışması olmayan özel gereksinimli öğrencilerin bu aşamadan başlamalarının hem yapabildikleri basamaklar itibari ile kendilerini başarılı hissetmelerini sağlamakta; hem de kodlama mantığının temelini daha iyi oturtmakta olduğu söylenebilir.

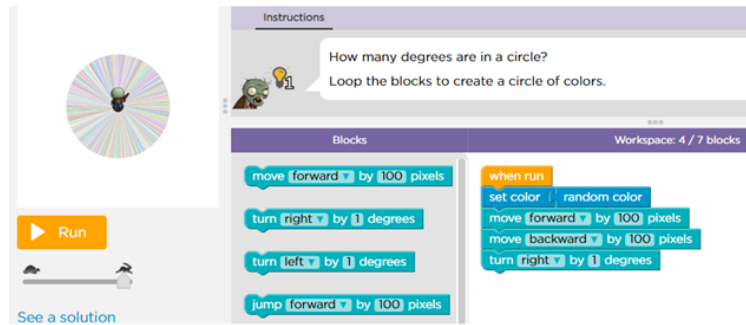
Ön koşul becerilerinin değerlendirilmesi sürecinde yön kavramlarına sahip olduğu gözlenmiş olmasına rağmen; uygulama sürecinde yer alan ve Şekil 5'te bir etkinlik örneği verilen labirent aşamasında öğrencinin yön kavramında sorunlar yaşadığı görülmüştür. Bu durumda araştırmacılar, bu önkoşul beceriyi öğrencinin bağımsız ve istenen düzeyde yapamadığı görüşünü benimsemişler ve çalışma kağıtları hazırlayarak ailesinin rehberliğinde çalışmasını istemişlerdir. Aile iş birliği ile bu konuda gelişme kaydedilmiştir.

Şekil 5*Katılımcının Zorlandığı Labirent Aşaması*

Öğrencinin en çok zaman harcadığı ve aynı zamanda motive olduğu aşamalar tasarım aşamaları olmuştur. Öğrenci bu aşamalarda önceki seviyelerde edindiği ilerletme, döngü, koşul gibi becerileri kullanarak oyun, diyalog, çizim gerçekleştirmiştir. Şekil 6'da öğrencinin tasarladığı çizimlerden bir örnek yer almaktadır.

Şekil 6*Katılımcının Code.org'da Bir Tasarımı*

Bu tasarımda öğrenci çok eğlendiğini ifade etmiştir (18.03.2019 tarihli görüntü kaydı). Araştırmacılar ürünleri incelediklerinde öğrencinin yazmış olduğu diyalogun öğrencinin iç sesini yansıtmış olabileceğini günlüklerine aktarmışlardır. Bir başka ifade ile katılımcının arkadaş edinmek istediğini, bu durumu tasarımda ifade etmek istediğini düşünmüşlerdir. Bir diğer bulgu olarak; öğrencinin en çok sevdiği etkinlik "Play Lab: Create a Story", sevmediği ve en çok zorlandığı etkinlik ise Şekil 7'de görülen Kurs 2, "Ders 11: Sanatçı: Döngüler" etkinliği olarak belirlenmiştir. Çemberin çevresini hesaplama, karaktere çemberin açısına göre yönerge verme gibi aşamaları olan seviyelerde oldukça zorlanmıştır.

Şekil 7*Katılımcının Zorlandığı Bir Aşama***OSB Olan Bir Öğrenciye Blok Tabanlı Kodlama Öğretim Sürecinde Karşılaşılan Problemler Nelerdir? Nasıl Çözülmemiştir?**

Bu başlık altında; öğrencinin davranış problemleri, kodlama öğretiminde kullanılan araçtan kaynaklı problemler ve Code.org platformunun kullanımında karşılaşılan problemler olmak üzere araştırma sürecinde karşılaşılan üç temel sorun yer almaktadır.

Öğrencinin davranış problemleri sürekli ve dersi sabote eder hale geldiğinde araştırmacılar değerlendirme toplantısında birtakım uygulama kurallarının konulması gerektiğine karar vermişlerdir. Bu karar üzerine kurallar oluşturularak öğrenci ile izlerlik sözleşmesi imzalanmış, kurallara uyduğu takdirde uygulama yapabilecekleri belirtilmiştir. Bir başka karşılaşılan problem, öğrencinin bilgisayarsız etkinliklerdeki motivasyon düşüklüğü olmuştur. Bilgisayarsız etkinlikler başka bir masada gerçekleştirilse de öğrenci sürekli bilgisayar ekranına bakmak, bilgisayara dokunmaya çalışmak gibi davranış problemleri göstermiştir. Bu durumda; bilgisayarsız etkinlikleri mümkün olduğunca dersin başında yapmak, mümkün değilse bilgisayarsız etkinlik sırasında bilgisayarın monitörünü kapamak ve bilgisayarsız etkinliğe olan ilgisinin sözel pekiştiricilerle pekiştirilmesi şeklinde çözüme gidilmiştir.

Kodlama öğretiminde kullanılan araçtan kaynaklı problemler ise öğrencinin kodlamayı masa üstü bilgisayarda değil, tablette yapmak istemesi ile ortaya çıkmıştır. Tablette kodlamayı gerçekleştirmek öğrenci için zorlayıcı olmuştur. Tablette çift parmak kullanımı, diğer elinin ekrana değmesi, tıkladığında istediği sekmeye girememesi (ekranın algılanmaması), yanlış sekmeye girme gibi sorunlarla karşılaşmıştır. Ayrıca; tablet ekranına uzun kod yazıldığında bir bütün olarak görülebilmesi, kaydırma çubuğu ile aşağı yukarı yapılması da bir sorun olmuştur (07.03.2019 tarihli görüntü kaydı). Bu gibi sorunlarla karşılaşan araştırmacılar öğrencinin masaüstünde daha başarılı olduğunu açıklamışlar, araştırmada tablet kullanımına son vermişlerdir.

Code.org platformunda karşılaşılan problemler; bazı aşamaların İngilizce temelli alıştırmalarının olması ve araştırmacıların dilinde çevirisinin olmamasıdır. Ders öncesi hazırlıklarda bu durum dikkate alınarak ekranların çevirileri yapılmış, ekran üzerine katılımcının dilinde çeviriler postitlere yazılarak yapılandırılmıştır. Bazı derslerdeki etkinlikler İngilizce puzzle etkinlikleri içerdiğinden o dersler öğrencinin dilinde işlenememiştir. Ayrıca derslerin öğretmenler için yol gösterici videolarının İngilizce olmasının; uygulamalarında Code.org kullanmak isteyen ancak İngilizce bilmeyen öğretmenleri sınırlandırabileceğini göstermektedir.

Blok Kodlama Öğretiminin OSB Olan Öğrenciye Ne Gibi Katkıları Olmuştur?

Kodlama öğretiminin öğrenci için hedeflenen ve hedeflenemeyen katkılar sunduğu belirlenmiştir. Bu katkılar Tablo 3'te sunulmuştur.

Tablo 3

Kodlama Öğretiminin Öğrenciye Katkıları

Alt temalar	Kategoriler
Araştırmanın beklenen katkıları	Problem çözme becerilerini geliştirme Kendini yönetme becerilerini geliştirme Üretimde bulunmaya motive etme İngilizce kelime dağarcığının artışı
Araştırmanın beklenmeyen katkıları	Bilgisayar becerilerinde gelişme (ör. ekran görüntüsü alma, çıktı alma) Sosyal becerilerde gelişme (ör. selamlaşma, iletişim başlatma) Ses hassasiyetinde azalma

Tartışma

Bu araştırmanın amacı OSB olan bir öğrenciye kodlama öğretimi sürecinin ayrıntılı bir şekilde aktararak; öğrencinin kodlamada elde ettiği kazanımların, süreçte karşılaşılan problemlerin ve çözüm önerilerinin incelenmesidir. OSB olan bir öğrenci için doğrudan öğretim yöntemi ile planlanan kodlama becerisi öğretim oturumları, çalışmanın öğrenci açısından kazanımları, karşılaşılan problemler ve çözüm önerileri çeşitli veriler analiz edilerek ortaya konmuştur. Bulgular, kodlama öğretimi sürecinin sonucunda öğrencide önemli kazanımlar elde edildiğini göstermiştir. Öğrenci, Code.org platformunda birinci ve ikinci düzeyi, araştırmacı gözetiminde verilen yardım düzeyi azaltılarak başarıyla bitirmiştir. İkinci düzeyinin son iki dersini hiç yardım almadan tamamlamış ve kodlama becerileri konusunda bağımsızlaşmıştır. Özel gereksinimli bireylere bilgisayarda kodlama öğretimi sürecinde tipik gelişen çocuklara uygulanan yapılandırıcı yaklaşımdan ziyade doğrudan öğretim kullanılmasının daha iyi bir başlangıç olduğu belirtilmektedir (Basham & Marino, 2013; Israel, Wherfel vd., 2015; Taylor, 2018; Taylor vd., 2017). Bu açıdan çalışmamızda doğrudan öğretimle başlayarak yardımın giderek azaltılması ve öğrencinin bağımsızlaşması bulgusu alan yazındaki çalışmaların bulgularıyla tutarlıdır (Taylor vd., 2017). Ayrıca bu çalışmada öğrencideki yardım düzeyini öğrencinin ihtiyaçları doğrultusunda vererek öğrencinin kendi yapabildiklerini görmesi, öz güven geliştirmesi ve ihtiyacı olduğunda yardım istemeyi öğrenmesi gibi ek kazanımlar da elde etmiştir.

Bu araştırma katılımcının teknolojiye olan yoğun ilgisinin kodlamaya yönlendirilmesiyle katılımcının ileriye dönük kariyer planlamasına katkıda bulunduğu görülmüştür. Katılımcı "Ben kodlamacı, yazılım

mühendisi olacağım.” şeklinde ifadelerde bulunmuştur. Bu durumda, STEM kapsamında yapılan eğitimlerin bireylerin kariyer planlaması için önemli katkılar sunduğu görüşünü desteklemektedir (Israel, Wherfel vd., 2015; Kalelioğlu, 2015; Knight, Wright, Wilson vd., 2019; Lee vd., 2020; Shattuck vd., 2014; Wei vd., 2013). Bu çalışma, katılımcının ilgi alanı doğrultusunda kariyer planı yapmasına olanak sağlamıştır. Ayrıca blok tabanlı kodlama platformunun hem eğlenceli hem de temel programlama mantığıyla öğrencilerin ileri düzeydeki bilgi işlemsel düşünme ve programlama becerilerini edinmelerinde temel oluşturduğu belirtilmektedir (Israel, Pearson vd., 2015). Alanyazında Code.org gibi platformlarda blok tabanlı kodlama uygulamaları ile tipik gelişim gösteren çocukların programlama becerilerini öğrendikleri, bu becerileri edinmeye ilişkin motivasyonlarının arttığı belirtilmektedir (Barradas vd., 2020; Lambić vd., 2020; Munoz vd., 2018). Bu süreçte OSB olan katılımcının da temel düzey alıştırmalardan, tasarım sürecine kadar gerçekleşen tüm aşamalarda eğlendiği ve motivasyonunun üst düzeyde olduğu görülmüştür.

Araştırmada hedeflenmeyen ancak süreçte öğrencide olumlu gelişmelerin görüldüğü durumlar tespit edilmiştir. Katılımcının ses hassasiyeti hem araştırmacılar tarafından gözlenmiş hem de ailesi ve öğretmeni tarafından belirtilmiştir. Araştırmanın başında bilgisayarsız etkinlik videolarını izlerken sesin açılmasına izin vermemesine rağmen, tasarımın içerisine ses ekleme gibi etkinliklerin yer aldığı aşamalarda sesi açarak, sesleri dinlediği gözlenmiştir. Bu durum araştırmanın öğrenciye sağladığı bir katkı olarak ifade edilebilir. Bir başka katkısı ise öğrencideki İngilizce kelime dağarcığının artışı ve İngilizceye olan ilgisinin artmasıdır. Google Translate platformunu kullanarak yardım almadan komutları adlandırmaya çalışması, araştırmanın öğrencideki problem çözüme ve bağımsızlaşma becerilerine katkı sunduğunu göstermektedir. Alan yazında bilgi işlemsel düşünme çalışmalarının en önemli katkılarından biri olan problem çözüme becerilerinin bu durumda gözlendiğini söylemek mümkündür (Elsahawy vd., 2020; Geist, 2016; Israel, Pearson vd., 2015; Israel, Wherfel vd., 2015; Kale & Yuan, 2021; Kalelioğlu, 2015; Taylor, 2018; Taylor vd., 2017).

Araştırmada elde edilen bir diğer bulgu ise doğrudan öğretim yöntemi ile desenlenen öğretim sürecinde öğretmende tablet bilgisayar, öğrencide masa üstü bilgisayar ile kodlama aşamalarının gerçekleştirilmesinin öğrenci motivasyonunu arttırmasıdır. Bu durum öğrenciye bir basamağı iki kez izleme fırsatı sunmuştur. Öğrenci bir öğretim oturumunda öğretmenin model olduğu tablet bilgisayar üzerinde uygulamalarını yapmak istemiştir. Tablet bilgisayar üzerinde yaptığı kodlama etkinliğinde farklı yerlere tıklaması, kod satırının hepsinin aynı sayfada görünmemesi gibi sorunlar ortaya çıkmış, öğrencinin motivasyonunun düşmesine neden olmuştur. Böyle bir beceri öğretiminde öğrencinin fare tutma, fare ile komut verme konusunda sınırlılığı yok ise masa üstü bilgisayarda yapılmasının daha uygun olduğu belirlenmiştir. Fare kullanımı da özel gereksinimli öğrenciler için bazen güç bir beceri olabilmektedir. Ayrıca bilgisayar kullanma konusunda yeterliliği olmayan öğrenciler için de sınırlılık yaratılmaktadır (Israel, Pearson vd., 2015). Bu bulgudan yola çıkarak ön koşul becerilerinin değerlendirilmesi ve özellikle bu becerilerin test edilmesi kodlama öğretimi için önemli bir aşama olduğunu söylemek mümkündür.

Katılımcı sadece programlama konusunda beceriler edinmemiş; bunun yanında kaydettiği belgeyi bilgisayarda bulma, ekran görüntüsü alma, yazıcıdan çıktı alma gibi daha önce sahip olmadığı beceriler de edinmiştir. Alan yazında da programlama öğretimin sadece mühendislik alanında kariyer edinimine katkı sunmadığı, günlük yaşamda bu becerilerin kullanılacağı veya başka meslekler edinilse de bu becerilere ihtiyaç duyulacağı belirtilmektedir (Arfe vd., 2020; Israel, Wherfel vd., 2015; Wright vd., 2020). Böylece katılımcımız bu öğretim sırasında ileride kullanabileceği becerileri de dağarcığına katmıştır.

Gerçekleştirilen bu çalışmada OSB olan öğrencinin blok tabanlı kodlama platformunda başarılı olduğu görülmüştür. Knight, Wright, Wilson ve diğerlerinin (2019) de belirttiği üzere OSB olan bireylerin bilgisayarda blok tabanlı kodlama konusunda başarılı oldukları görülmektedir. Bu konuda daha çok çalışmaya ihtiyaç duyulmakta ve sadece kuramsal bazda değil, uygulamada yer alan öğretmenlerle özel gereksinimli öğrencilere nasıl bilgi işlemsel düşünme becerilerinin öğretileceği, hangi platformlardan hangi öğretim yöntemleri ile sunulacağı konusunda çalışmalar yapılması gerektiği görülmektedir.

Bu çalışma, teknolojiye ilgi duyan, OSB olan bir öğrenci ile gerçekleştirilmiştir. Elde edilen veriler bu çerçeveden değerlendirilmelidir. Kodlama için ön koşul becerileri sağlamayan öğrencilerin de öğretimine ilişkin çalışmalarına yer verilebilir. Kodlama becerilerinin öğretiminde bireylerin kazandığı bir beceri de iş birliği yapma, ortak çözümler üretmedir (Israel, Wherfel vd., 2015; Karp & Maloney, 2013). Bu çalışma birebir öğretim ortamında gerçekleştirildiğinden. Katılımcıya bu beceriler kazandırılmamıştır. İleriki araştırmalarda kodlama becerilerinde OSB olan öğrencilerin iş birliği yapma düzeylerindeki gelişmeler de incelenebilir.

Yazarların Katkı Düzeyleri

Arařtırmacılar arařtırmanın her ařamasında ortak olarak alıřmıřlardır. Birinci yazarın verilerin analizi ve alıřmanın raporlanmasında daha fazla katkısı olurken, ikinci yazar verilerin toplanması ařamasında daha fazla katkı sunmuřtur.

Teřekkür

Arařtırmamızın katılımcısı öđrencimize, öđrencimizin gerekleřtirdiđimiz uygulamalara katılımı konusunda bize yardımcı ve destek olan velisine, alıřmamıza deđerli fikirleriyle katkı sunan Dr. Öđr. Üyesi Onur İřbulan'a ve uygulamaların video kaydına alınmasında yardımcı olan sevgili öđrencimiz İrem Avlar'a teřekkürlerimizle.