Informatics in Education, 2024, Vol. 23, No. 1, 125–143

DOI: 10.15388/infedu.2024.02

The Effect of Creating 3D Objects with Block Codes on Spatial and Computational Thinking Skills

Mehmet KÜÇÜK[1,*], Tarık TALAN[2], Muhammet DEMIRBILEK[1]

[1]*Suleyman Demirel University, Isparta, Türkiye.*
[2]*Gaziantep Islam Science and Technology University, Gaziantep, Türkiye.*
e-mail: mail.mehmetkucuk@gmail.com, ttalan46@hotmail.com, mdbilek@gmail.com

Received: January 2023

Abstract. This study investigated the effects of 3D model building activities with block codes on students' spatial thinking and computational thinking skills. The study group consists of 5th grade students in a secondary school in the Central Anatolia region of Turkey. For the study, a pretest-posttest control group was utilized within the experimental design. A total of 66 students participated, 23 in the experimental group and 43 in the control group. While the activities prepared on the Tinkercad platform were applied in the experimental group, the courses were taught using the traditional teaching method in the control group. The study covers a period of three-weeks in the course information technologies and software. The study used the computational thinking levels scale and spatial thinking test scales as data collection instruments. The data was analyzed using both descriptive statistics and independent samples t-tests. Based on the study findings, there were no significant differences observed in the levels of computational thinking skills levels and spatial thinking test scores between the experimental and control groups.

Keywords: computational thinking, spatial thinking, block coding, tinkercad, 3D design.

1. Introduction

Especially in the last century, the development of technology has progressed much faster than in previous centuries, and it has been used for various purposes. However, individuals must have the skills of the particular age in order to utilize technology for a particular objective. The concept of 21st century skills, which has recently entered our lives, expresses what skills individuals should be equipped with to keep up with the rapidly changing and digitized times, to meet the expectations of the times, and to be successful. Skills students should acquire in the 21st century include self-management, communication, adaptability to teamwork, problem solving and critical thinking and information, media, and technology literacy (Battelle for Kids, 2019). Wing (2006), on the other

* Corresponding author

hand, states that computational thinking skills are among the 21st century skills. Computational thinking is an essential skill that everyone should acquire in order to adapt to these times (Engelhardt, Chioccariellove & Ferrari, 2018; Grover, 2018).

Some methods are used to enable individuals to acquire computational thinking skills. One of them is to gain this skill with the help of coding. Computer programs can be used for coding. In general, these programs are built on block coding and are 2D (two-dimensional). On the other hand, the number of programs and platforms with 3D (three-dimensional) content and where block coding can be done is quite low. One of them is the Tinkercad (`www.tinkercad.com`) site used in the current study. Tinkercad differs from others in that it can create 3D models with block codes. Developed by Autodesk, Tinkercad is a web-based platform where 3D designs, circuit design and code blocks can be done and shared. In the 3D designs menu, 3D model studies can be done. In circuit designs, electronic circuit designs and Arduino circuit designs can be made and controlled by simulation. In addition, the Arduino circuit board can be coded by block or text. In the code blocks section, 3D objects can be designed with block coding. In order to create an object here, all sub-parts must be created with separate encodings and then combined. In the current study, 3D model creation activities were carried out with block codes using the code blocks section of Tinkercad.

3D modeling is important for mental design and visualization. Although advanced computers and software are accessible today to create 3D models, it is necessary to develop visualization skills in mental processes such as imagining, specifying and creating intricate designs within the three spatial dimensions (X, Y and Z). Developing students' visualization skills is essential to develop their design skills in many fields such as engineering, architecture, graphic design (Mcclurg *et al.*, 1997; Yıldız, 2009).

Spatial thinking skill is defined as being able to visualize the object in mind, to think of different points, to rotate and size it in space (Olkun, 2003). Spatial thinking skills contribute to the easier learning of subjects such as mathematics, geometry and physics, and have an impact on the success of these areas (Arslan Namlı & Aybek, 2022; Mcclurg *et al.*, 1997). In addition, Clements (1998) observed that children with improved spatial abilities were more successful in solving mathematical problems.

The use of 3D virtual environments instead of concrete objects in lessons improves students' spatial thinking skills (Yıldız, 2009). For example, a study involving young people aged 11–15 showed that the 3D modeling process helps develop spatial awareness skills (Šafhalter, Vukman & Glodez, 2015). Many of the 3D modeling tools startle people with problems such as dealing with geometry that is complex for beginners and difficult for drawing. Platforms like Tinkercad are pretty easy for beginners, with an easy interface, fewer foreign words, and simpler geometry (Kelly, 2014). With the new plugin of Tinkercad, 3D models can be made with simple block codes.

1.1. *Computational Thinking*

The concept of computational thinking, which is shown as one of the 21st-century skills (Grover, 2018), was originally introduced by Papert in 1996. Papert argues that computational ideas contribute to the development of learning and thinking. Wing (2006), who brought the concept of computational thinking back to the agenda in 2006, stated that this concept is among the basic skills of all fields, not just informatics. The reason why computational thinking creates a new type of inquiry in all disciplines is that it has a field of application for all disciplines, not just computer science (Bundy, 2007; Denning, 2009). Denning (2009), who also argues that computational thinking skills cover all disciplines, states that acquiring this concept has nothing to do with computer science, it was used as algorithmic thinking in the analysis of problems in the 1950s, therefore the basis of computational thinking is algorithmic thinking.

Kafai and Burke (2013) thought that seeing computational thinking skills as individuality would be a deficiency, and they should be social rather than individual. The easiest way to achieve social gain is through school curricula. When computational thinking is integrated with school curricula, it has been observed that the development of mathematical thinking skills increases (Oluk, Korkmaz & Oluk, 2018).

Computational thinking aims to foster not just the decision to pursue training in computing careers but also the holistic development of citizenship within a free and democratic society (Caeli & Bundsgaard, 2020). Tedre and Denning (2016) express computational thinking as computational ideas and mental habits in the process of creating calculations made with software, simulation and machines in the field of information technologies. Csizmadia *et al.* (2015) defined computational thinking as a concept consisting of a logical reasoning process in which systems, processes and products can be better understood in order to better solve problems as a cognitive thinking process. These definitions share a fundamental focus on problem-solving processes, encompassing tasks such as problem understanding, problem formulation, and effective problem-solving strategies. Moreover, certain studies assert that computational thinking cannot be comprehensively defined without incorporating essential skills like creative thinking, algorithmic thinking, cooperative learning, critical thinking, communication, and problem solving. The aim is to underscore the significance of these skills in the context of computational thinking (ISTE, 2015; Korkmaz, Çakır & Özden, 2015). It is known that programming education has an important place in order to provide individuals with these skills (Akpınar & Altun, 2014; Shin, Park & Bae, 2013). Within this context, programming education can be regarded as a valuable means to acquire computational thinking skills (Lye & Koh, 2014; Oluk, Korkmaz & Oluk, 2018). Moreover, numerous countries have begun incorporating computer programming further into their educational curricula, aiming to enhance students' computational thinking abilities (Bargury *et al.*, 2012; Jones *et al.*, 2014; Kalelioğlu & Gülbahar, 2014).

1.2. *Spatial Thinking*

Many studies and definitions have been made on spatial thinking since the 1950s. According to Carroll (1993), spatial thinking is defined as perception, visual association, imagination and interpretation, while Olkun (2003) defined spatial thinking as the ability to create, animate and make sense of the object in two or three dimensions, as a whole or divided into parts. Spatial thinking involves a variety of cognitive skills that enable individuals to organize, reason about, and mentally manipulate both real and imaginary spaces. These skills encompass the ability to reason about the size, shape, direction, orientation, and trajectory of objects, visualizing relationships between objects, and/or these relationships mentally, and reasoning about the relationships of objects and their spatial and temporal dimensions (Gagnier, Holochwost & Fisher, 2022).

Spatial thinking is one of the important topics that has attracted attention in science, mathematics, engineers, architecture, astronomy and many other fields, especially in the last two decades. Additionally, it is observed in the literature that studies on the development and evaluation of this skill in many different fields continue to increase (Yurt, 2011). Studies have indicated that spatial thinking can be developed with education, experience, appropriate tools and activities (Olkun, 2003; Rafi, Samsudin & Ismail, 2006; Yurt, 2011). Again, researchers argue that the presence of a suitable instructional setting plays a crucial role in enhancing spatial skills (Alyeşil Kabakçı & Demirkapı, 2016; Demirkaya & Masal, 2017; Özcan, Akbay & Karakuş, 2016). However, it can be said that in which direction and how this development will take place is still open to debate.

1.3. *Coding*

Coding skill has taken its place among the basic requirements of the digital world, such as literacy and mathematics, which are among the basic requirements of today's world (Uzunboylar, 2017). On the other hand, Somuncu (2021) states that coding is a skill and includes problem solving processes. Coding skills are important for individuals to develop new software by producing their own solutions instead of using ready-made software for the problems they encounter (Tağci, 2019). While gaining coding skills, children can develop basic skills in many areas such as evaluating, planning, problem solving, teamwork and different communication skills (Uzunboylar, 2017).

Providing coding education to children at a young age contributes positively to the development of computational thinking skills (Portelance, 2015). In addition, it is thought that giving coding experiences in fun and simpler ways at an early age will contribute to the ability to learn complex programming languages more easily and gain reasoning skills in later ages (Scharf, Winkler & Herczeg, 2008).

In the 21st century, the technology sector has an important place in the fields of economy and trade. Many countries start coding education at an early age with the thought that digital development will be effective in the field of economy. Many countries, realizing that coding education given to children brings success especially in computational thinking, creativity and problem solving, have started to include coding education in their curricula starting from primary school and even kindergarten (Baz, 2018). It can be said that one of the aims of coding education given to children at an early age is to gain these skills. On the other hand, Turkey has made the information technologies course a compulsory course in the 5th and 6th grades. It aims to gain some targeted skills by adding the unit "Problem solving and algorithm" to the curriculum (Pakman, 2018).

1.4. *Block-Based Coding*

Today, with the rapidly developing technology, new generations should receive computer science education at an early age in order for the society to be a producer rather than a consumer of technology. Giving coding education at an early age is also supported by many educational philosophies (Çavdar, 2018). In order for coding education to appeal to a wide age range, it should be understandable and easier to learn. Considering coding structures today, they can be classified as text-based and block-based.

Like foreign language learning, computer programs are initially complex and take time to learn. Those who are just starting to learn coding give up because they experience a complex process without understanding the basics of coding and the logic of algorithms (Akyol Altun, 2018). Block-based coding applications have been developed in order to gain coding skills more easily. Unlike text-based programming languages, block-based coding platforms like Scratch, Code.org, MakeCode, or Snap allow programs to be developed by combining blocks based on logical rules, similar to assembling Lego pieces or solving a jigsaw puzzle (Haladjian, Bredies & Brügge, 2016; Toma, 2023).

Block-based coding applications have been created to reach more users and include multimedia tools such as pictures and music, attracting students' attention and increasing their motivation (Pakman, 2018). The use of multimedia tools such as visual and auditory is the concretization of the intangible programming concepts such as variables, loops and conditions, which are difficult to perceive, by supporting them with animations (Erol, 2015). In these environments, more sense organs are included to facilitate permanent learning. In this way, it increases its interest and makes learning fun.

For those new to coding, the block-based coding environment is simpler than text-based coding due to the created design and ease of understanding (Weintrop & Wilensky, 2015). Block-based coding applications are more interesting for new learners, as errors such as compilation or syntax are common in text-based programming.

It can be said that there has been a notable rise in research on block-based coding in recent years. As an example, Totan and Korucu (2023) conducted a study examining the impact of block-based coding education on students' computational thinking skills as well as their attitudes towards coding. The study utilized the Blocky programming environment. The findings of the research revealed that the block-based coding environment with Blocky significantly influenced students' self-efficacy perception, computational thinking skills, and attitude towards learning coding. The observed effect size was large. Again, in the study conducted by Dilmen, Kert and Uğraş (2023), the usability of code.org was examined in terms of efficiency, efficacy, and user satisfaction. The research findings revealed that all children were able to complete the tasks assigned to them; however, they encountered difficulties during the block searching process. On the other hand, Talan (2020) has comprehensively examined the studies on the use of the Scratch program. According to the results of the analysis, it has been determined that the use of the Scratch program in education has positive effects on higher-order thinking, attitude, motivation, self-efficacy and academic achievement.

On the other hand, Arslan Namli and Aybek (2022) examined the impact of block-based programming and unplugged coding teaching activities on students' computational thinking skills, self-efficacy, and academic achievement. The study utilized unplugged coding activities for the experiment-1 group, while block-based programming activities were employed for the experiment-2 group. In the control group, traditional teaching practices were implemented. The research findings revealed no significant difference in computational thinking self-efficacy scale scores among the groups. However, it was observed that experiment-2 had higher scores on the International Informatics and Computational Thinking Activity Task Test and the Computational Thinking Performance Test compared to experiment-1 and the control groups.

In their study, Scullard *et al.* (2019) examined the development of computational thinking skills of non-computer science students with Scratch program activities. As a result of the study, it was observed that there was a significant increase in the computational thinking skills of students whose Scratch program activities were not computer science majors. In addition, it was determined that the activities of the Scratch program contributed to the development of logical thinking and abstraction skills in students.

Aydoğdu (2020) examined the influence of block-based programming on the development of computational thinking skills and self-efficacy perceptions. For this, a group of pre-service teachers conducted block-based coding activities for four-weeks. According to the data obtained as a result of the activities, it was determined that block-based coding had no effect on the computational thinking skills of pre-service teachers, but positively affected the self-efficacy perception towards programming.

Timur *et al.* (2021) conducted a four-week training program for pre-service teachers, utilizing the Scratch application. The researchers employed an interview form to assess the participants' experience with block-based coding in education. The research findings indicated that the most of the participants shared the belief that block-based coding education should began at a young age, similar to language education, and that all students should have access to this type of education. Furthermore, the participants highlighted that block-based coding enhances students' creativity and contributes to the development of both tangible and abstract materials.

In their research, Chaudhary, Agrawal, and Sureka (2016) aimed to analyze the approaches to the problem and the alternative solutions produced by using the Lego Mindstorms EV3 kit with nine primary school students. Students were given training on visual programming concepts and Lego components, enabling the programs prepared in the visual environment to interact with the hardware. Afterwards, the students were asked to create a set of applications and observations were made during this process. Then, the data were collected by asking questions to the students. According to the results obtained from the collected data, it has been seen that block-based coding programs are more effective in terms of skills gained than text-based programming and contribute to the development of students' skills such as engineering, problem solving, computational thinking, robot programming and collaborative learning.

1.5. *Purpose of the Study*

Computational thinking skill is one of the skills expected to be acquired today, when considered together with its basic sub-fields (Grover, 2018). These gains are generally tried to be gained in the information technologies and software course at school. There are different methods for this. For example, Weinberg (2013) stated that there are three different ways to add computational thinking skills to the curriculum. The first of these is the computer-free learning environment. This can include mental games, card games, mind games, and various physical activities. The second is the game and robot programming environment, that is, programming the robot hardware by gamification. The third is initial learning environments. These are simplified and visually enhanced game design and programming activities. These activities start with block coding. Here, block coding and visual design programs such as Code.org, Scratch and Blockly are used as tools. In the current study, the block coding method was chosen because spatial thinking skills would also be gained. There are many software and applications available for block coding. However, in such applications, processing is usually performed on 2D. In the current study, since it is aimed to provide students with spatial thinking skills, 3D modeling with block coding was used and the Tinkercad site, which can gain these two skills together, was used.

3D modeling helps develop spatial skills (Yıldız, 2009). Factors such as age and experience are important in the development of spatial skills, but previous research has also found that this skill can be developed with the help of 3D modeling. An illustration of this can be seen in a study where the utilization of 3D modeling was found to enhance spatial visualization skills among both engineering students and students engaged in graphical studies (Devon *et al.*, 1994).

In the studies, gamified coding applications are used for the acquisition of students' computational thinking skills. Studies have shown that students' dealing with such applications increases their computational thinking skills (Resnick *et al.*, 2009). But these applications are generally based on moving 2D or 3D objects. In the current study, activities are carried out on the creation and animation of 3D objects with block coding, which is one of the simplest visualized forms of coding. In the study, the activities created with Tinkercad enable students to create a 3D object by coding for the first time and to move this object. It is assumed that with these activities, both computational thinking and spatial thinking skills of the student will increase. It is important to examine these two concepts together, which have an important place among the 21st century skills, and to reveal their results, in terms of guiding researchers who will work in the related field.

Upon reviewing the literature, it becomes apparent that numerous studies have been conducted on computational thinking, coding, spatial thinking, and 3D objects. The number of researches, especially in the fields of computational thinking and coding, has recently increased significantly. Considerable research has been conducted regarding the topic of spatial thinking and 3D objects. Nevertheless, upon analyzing the literature, it has been identified that there is a scarcity of studies focusing on the acquisition of both computational thinking and spatial thinking skills through the same application. Studies show that it is important to examine these two concepts together (Engelhardt, Chioccariello & Ferrari, 2018; Grover, 2018; Mcclurg *et al.*, 1997; Yıldız, 2009). However, it is thought that such studies that investigate the relationship between spatial and computational thinking concepts and block coding education, especially with the increasing prevalence of coding education in many countries, can contribute to the literature.

In this study, activities were carried out on the creation of 3D objects by block coding in the information technologies and software course of secondary school students. Along with the activities, it was examined whether the students' computational thinking and spatial thinking skills developed or not. In this context, the current study aimed to examine the effect of creating 3D objects with block codes on students' spatial and computational thinking skills. In accordance with the study's objective, an experimental group and a control group were established. The group in which the lessons were taught by creating 3D objects with block codes was called the experimental group, and the group in which the lessons were taught in the traditional face-to-face learning environment was called the control group. In accordance with the overall objective of the study, the researchers aimed to find answers to the following sub-problems:

1. Do the computational thinking skills pre-test scores of the control and experimental groups differ significantly?
2. Do the spatial thinking pre-test scores of the control and experimental groups differ significantly?
3. Do the computational thinking skills post-test scores of the control and experimental groups differ significantly?
4. Do the spatial thinking post-test scores of the control and experimental groups differ significantly?

2. Method

2.1. *Study Design*

The current study adopts a quantitative approach and utilizes an experimental design with a pretest-posttest control group. Trial models refer to research models in which the desired data to be observed is generated in order to attempt to determine the cause-effect relationship under the control of the researcher. The pretest-posttest control group model consists of two groups. One group is assigned as the experimental group, while the other group functions as the control group. In both groups, measurements are made before and after the experiment (Köse, 2010). While models were developed using the Tinkercad site in the experimental group during the research process, the traditional face-to-face learning method was used in the control group.

2.2. *Study Group*

The study group for this research comprises 5th grade students who are attending a secondary school in the Central Anatolia Region of Turkey. There were 43 students in the experimental group, 25 (58.1%) female and 18 (41.9%) male. The control group consisted of a total of 23 students, 8 (34.8%) female and 15 (65.2%) male.

2.3. *Application Steps of the Study*

Prior to commencing the application, consent for research was acquired from the Ministry of National Education. Course contents were prepared and planned. A three-week course plan was created by taking into account the objectives, achievements and activities. Before the application, the researcher informed the teacher of the lesson about the applications to be made on the Tinkercad site. After the control and experimental groups were determined, the first lesson pre-test data collection tools were applied. Before and after the application, two course hours (one-hour pre-test, one-hour post-test) were allocated to collect data and inform about the study. The study was completed in a total of three-weeks (6 course hours). In the application, a total of four in-class activities were carried out. In the last lesson of the application process, post-test data collection tools were applied to the control and experimental groups. Students who did not regularly participate in the lessons and studies conducted during the research or who could not apply pre-test or post-test data collection tools were not included in the scope of the study. The educational activities determined for the control and experimental groups were carried out in the same semester and by the same teacher.

The activities prepared within the research scope were applied by the students in the experimental group, considering the development of computational thinking and spatial thinking skills. The Tinkercad site was used for the events. A tutorial was developed to guide the execution of the course activities. At the beginning of the course, the students were able to ask their teachers what they did not understand about the previous topics. While preparing the activities, the level of the students, the level of complexity and the difficulty of the subjects were taken into consideration. The acquisition of computational thinking sub-dimensions and the acquisition of spatial thinking sub-dimensions were taken into account and four activities were prepared. These activities are the object creation and movement activity, the punched box activity, the goal and ball activity, and the table creation activity, respectively. During the activity, each student implemented the activity from their own computer and shared ideas among each other. The teacher provided the students with step-by-step instructions, informing them about the activities to be carried out within the classroom. Subsequently, the students were instructed to sequentially perform the activities. Throughout the process, the teacher provided guidance and assistance to the students. The teacher intervened at the points that were not understood and the misunderstandings were corrected. In contrast, the control group of students received instruction using the traditional teaching approach.

The activities have been meticulously developed based on expert opinions. In accordance with these expert perspectives, each activity was carefully crafted to encompass aspects of creativity and algorithmic thinking skills, which serve as sub-dimensions of computational thinking. Also, the punched box activity was specifically crafted to foster problem-solving skills. Moreover, the goal and ball activity was developed to facilitate the acquisition of both cooperative and critical thinking skills. Lastly, the table creation activity aimed to enhance both cooperation and problem-solving skills. Furthermore, all the activities were meticulously designed to encompass the development of spatial thinking skills.

2.4. *Data Collection Instruments*

2.4.1. *Computational Thinking Levels Scale*

The study employed the computational thinking skill levels scale, developed by Korkmaz, Çakır, and Özden (2015), to assess the levels of computational thinking skills among the students. This scale comprises five factors, namely creativity, cooperativity, algorithmic thinking, critical thinking, and problem-solving skills. The reliability coefficient (Cronbach Alpha value) of the scale was found to be 0.80. The scale consists of 22 items and is in a five-point Likert type. Korkmaz, Çakır, and Özden (2015) emphasize that raw scores obtained from this scale can be converted into standard scores between 20–100 points and this does not depend on the applied groups. Scores between 20–51 correspond to low level, those between 52–67 correspond to medium level, and those between 68–100 correspond to high level.

2.4.2. *Spatial Thinking Test*

In the study, the spatial thinking test developed by Yurt (2011) was used to measure the spatial intelligence levels of the students. There are 16 items in this test. In the test, when objects made of 3D cubes are viewed from different angles, they are asked to visualize their invisible parts. In each question, the number of cubes for these objects is asked. The developed test was applied to 211 students, including 6^{th}, 7^{th} and 8^{th} grade students, in order to obtain gains such as perceiving 3D structures correctly and comparing the parts that make up 3D structures with each other. Yurt (2011) determined the reliability coefficient (Cronbach's Alpha value) as 0.92 together with the items with high discriminatory power.

2.5. *Data Analysis*

The data obtained in the study were subjected to analysis using the SPSS 18.0 program. All objectives of the study were tested and interpreted at a confidence level of 0.95 ($p = 0.05$). Before the analysis, Z score conversion was performed to see if there were any outliers in the data and it was determined that there were no outliers. Then, frequency (f), arithmetic mean (X), standard deviation (Ss), minimum (Min), maximum

(Max), skewness (TC) and kurtosis coefficients (BK) were calculated and the assumptions required for the analysis were examined.

In the data analysis of the study, the t-test, which is one of the parametric tests, was utilized. The main assumptions for using parametric statistics are that the data obtained are intermittent, homogeneous or proportional (Kalaycı, 2010). A Levene test was conducted to evaluate the homogeneity of variances in the data obtained from the study groups and it was observed that the variances showed homogeneity ($p > 0.05$). Following the confirmation of homogeneity in variances, an independent samples t-test was conducted to examine the presence of a significant difference between the averages of scores obtained from the experimental and control groups. This test is employed to ascertain if there exists a significant difference between the means of two independent samples, or to compare the group values on the continuous variable (Seçer, 2013).

3. Results

3.1. *Comparison of the Pre-test Results of the Groups*

Prior to the experimental application, a pre-test was conducted on the groups in order to assess the similarity between the experimental and control groups and to evaluate the efficacy of the methods. Table 1 presents the results of the pre-test for the computational thinking skill scale.

Upon examining Table 1, it is seen that the arithmetic mean scores of the experimental group in the pre-test of the computational thinking skill scale are (X = 68.62) and the control group (X = 67.78). As a result of the analysis, it was seen that the pre-test computational thinking skills levels of the control and experimental groups were

Table 1
Pre-test analysis results of the scale of computational thinking skill levels of the groups

Factors	Group	N	X	Ss	Min.–Max.	Sd	t	p
Creativity	Experiment	43	14.35	3.12	3.64–18.18	64	0.30	0.76
	Control	23	14.11	3.17				
Algorithmic Thinking	Experiment	43	13.42	2.91	3.64–18.18	64	0.08	0.93
	Control	23	13.35	2.91				
Cooperativity	Experiment	43	13.67	3.85	3.64–18.18	64	-0.36	0.71
	Control	23	14.03	3.65				
Critical Thinking	Experiment	43	12.70	3.95	3.64–18.18	64	-0.14	0.88
	Control	23	12.84	3.21				
Problem solving	Experiment	43	14.46	5.32	5.44–27.28	64	0.80	0.42
	Control	23	13.43	3.97				
Total	Experiment	43	68.62	9.58	20–100	64	0.36	0.71
	Control	23	67.78	7.71				

Table 2

Pre-test analysis results of the spatial thinking test of the groups

Group	N	X	Ss	Min.–Max.	Sd	t	p
Experiment	43	6.27	3.20	0–16	64	-1.61	0.11
Control	23	7.69	3.74				

close to each other and did not create a statistically significant difference ($t_{(64)} = 0.36$; $p > 0.05$).

The spatial thinking test was also administered to both the experimental and control groups prior to the application. The findings regarding the pre-test results of the spatial thinking skill scale are displayed in Table 2.

According to Table 2, the arithmetic mean of the scores of the experimental group from the pre-test of the spatial thinking test is (X = 6.27) and the arithmetic mean of the control group is (X = 7.69). As a result of the analysis, it was seen that the pre-test spatial thinking skills levels of the control and experimental groups were close to each other and did not create a statistically significant difference ($t_{(64)} = -1.61$; $p > 0.05$).

3.2. *Comparison of the Post-test Results of the Groups*

The study aimed to evaluate the impact of the intervention conducted after the application process by analyzing the scores of the students in the study group on the post-test assessments of computational thinking and spatial thinking skills. Table 3 provides the results of the post-test assessments for the computational thinking skill scale across the groups.

Upon reviewing Table 3, it can be observed that the experimental group's arithmetic mean for the post-test of the computational thinking skill scale is (X = 66.59), while the control group's arithmetic mean is (X = 64.90). The analysis revealed that the post-test scores of the groups on the computational thinking skill scale were similar and did not exhibit any significant difference ($t_{(64)} = 0.68$; $p > 0.05$). This difference does not mean a significant difference in creativity ($t_{(64)} = 0.14$; $p > 0.05$), algorithmic thinking ($t_{(64)} = -0.75$; $p > 0.05$), cooperativity ($t_{(64)} = 0.20$; $p > 0.05$), critical thinking ($t_{(64)} = 0.43$; $p > 0.05$) and problem solving ($t_{(64)} = 1.31$; $p > 0.05$) dimensions.

In the study, a t-test was also conducted to determine whether the spatial thinking skills of the groups differed significantly, and the results are given in Table 4.

According to Table 4, the arithmetic mean of the scores obtained from the post-test of the spatial thinking test of the experimental group is (X = 9.53) and that of the control group (X = 8.43). The analysis reveals that there is a close similarity between the post-test scores of the groups on the spatial thinking skill scale, indicating no significant difference ($t_{(64)} = 1.22$; $p > 0.05$). Upon reviewing the analysis results, it becomes evident that there is no significant difference between the experimental and control groups in the post-test of the spatial thinking scale.

Table 3
The results of the post-test analysis for the levels of computational thinking skills across the groups

Factors	Group	N	X	Ss	Min.–Max.	Sd	t	p
Creativity	Experiment	43	14.03	3.56	3.64–18.18	64	0.14	0.88
	Control	23	13.91	2.85				
Algorithmic Thinking	Experiment	43	12.09	3.48	3.64–18.18	64	-0.75	0.45
	Control	23	12.76	3.37				
Cooperativity	Experiment	43	12.98	3.68	3.64–18.18	64	0.20	0.83
	Control	23	12.76	4.55				
Critical Thinking	Experiment	43	12.43	3.94	3.64–18.18	64	0.43	0.66
	Control	23	12.01	3.23				
Problem solving	Experiment	43	15.05	4.92	5.44–27.28	64	1.31	0.19
	Control	23	13.43	4.44				
Total	Experiment	43	66.59	10.09	20–100	64	0.68	0.49
	Control	23	64.90	8.46				

Table 4
Post-test analysis results of the spatial thinking test of the groups

Group	N	X	Ss	Min.–Max.	Sd	t	p
Experiment	43	9.53	3.27	0–16	64	1.22	0.22
Control	23	8.43	3.84				

4. Discussion

In the study, the impact of 3D model creation activities with block codes on students' computational thinking and spatial thinking skills were examined. The study involved a total of 66 5th grade students, comprising both a control group and an experimental group. While the lessons were taught according to the traditional education curriculum in the control group, activities in which 3D objects were created with block codes were carried out in the experimental group. In these activities, 3D objects were designed and moved with visualized block coding in Tinkercad.

4.1. *Results on Computational Thinking Skills*

Within the scope of the study, it was concluded that there was no significant difference between the post-test scores of the computational thinking skill levels scale of the experimental and control groups. Upon examining the existing literature on the subject, it is evident that there is a higher prevalence of studies indicating a significant relationship between computational thinking and block coding (Atmatzidou & Demetriadis, 2016; Oluk, Korkmaz & Oluk, 2018; Totan & Korucu, 2023). On the other hand, certain studies exhibit congruity with the results of this study and no significant relationship can be found (Atman Uslu, Mumcu & Eğin, 2018; Yolcu, 2018). Webb Rosson (2013) points out that some of the reasons for not finding a significant relationship between computational thinking and coding in studies are the scarcity of resources used in the evaluation of computational thinking skills and the inability to measure different sub-skills. Kazakoff, Sullivan, and Bers (2013) stated that activities on this subject should be long-term in order to contribute to computational thinking skills. Atmatzidou and Demetriadis (2016) emphasized that since measuring computational thinking skills is a complex process, it would be appropriate to use more than one assessment tool.

In the current study, the reasons for not finding a significant relationship between computational thinking and applied activities can be shown as students' first use of Tinkercad, difficulties in creating 3D objects, data collection using a scale and method, and short implementation time. In addition, since the study was carried out in a private school, it is possible that the students received coding education at an earlier age or through out-of-school activities. This may have caused no significant difference between the two groups.

For the acquisition of computational thinking skills, the acquisition of sub-dimensions is required. Since sub-dimensions are high and each of them has a different depth, it is difficult to gain these skills with a single activity method, which may have caused the inability to find a significant difference between the two groups.

4.2. *Results on Spatial Thinking Skills*

The study's findings led to the conclusion that there is no significant difference between the post-test scores of the spatial thinking test for both the experimental and control groups. Upon examining the available literature on spatial thinking, it becomes evident that there are more studies that have a significant relationship with the use of virtual environments (Martín-Dorta, Saorín & Contero, 2008; Mcclurg *et al.*, 1997; Mumcu & Yıldız, 2015; Yurt, 2011). There are very few studies that do not find a significant relationship. For example, Çetin (2019) conducted a study exploring the correlation between spatial thinking and augmented reality, and the results did not reveal a significant relationship. In another study, Sarı (2012) investigated the relationship between spatial thinking and supported education utilizing concrete models. However, the study did not find a significant relationship. Sarı (2012) provided an explanation for this, suggesting that inadequate understanding of the concept of spatial thinking by the teacher and the students' familiarity with the traditional model contributed to easier communication with the teacher, thereby influencing the results.

In the current study, the reasons for not finding a meaningful relationship between spatial thinking and applied activities are that students are using tinkercad.com for the first time, students are creating 3D objects for the first time, the activities prepared cannot provide the desired qualities, the teacher cannot fully grasp the subject, and the application period is short. can be displayed.

4.3. *Limitations*

This study focuses exclusively on 5th grade students attending a secondary school located in the Central Anatolia Region of Turkey. The study is limited to a three-week activity conducted using the tinkercad.com website for the information technologies and software course. The acquisition of study findings is limited to assessing the students' levels of computational thinking skills and spatial thinking test.

4.4. *Suggestions*

In light of the study's findings, the following recommendations are proposed for researchers working in the field of computational thinking, spatial thinking, and block coding, aiming to facilitate their development and advancement:

- In the study, it was concluded that 3D object creation activities with the help of block codes on the Tinkercad did not make a statistically significant improvement in students' computational thinking and spatial thinking skills. However, the current study is limited to 66 students who continue their education in the 5th grade of a secondary school in the Central Anatolia Region of Turkey. Expanding the study with different levels and different participations in different regions will contribute to the literature.
- The present study was exclusively conducted within a private school. It is worth noting that a considerable number of students in private schools may have access to private training opportunities, influenced by their family's financial circumstances. Conversely, students attending public schools might not have the same level of access to such opportunities. Consequently, it would be beneficial to conduct a comparative study that includes both public and private schools to explore this aspect further.
- Taking into account the developmental characteristics of 6th grade students, it becomes evident that they are at the initial stage of transitioning from concrete thinking to abstract thinking (Senemoğlu, 2018). Conducting 3D modeling studies in a virtual environment for students in the 6th, 7th and 8th grades of primary education may yield diverse outcomes.
- Using quantitative and qualitative research methods together in spatial thinking and computational thinking studies can make the results more meaningful.
- Different results can be obtained if the activities have a longer duration of implementation.
- Considering the acquisitions of computational thinking and spatial thinking skills, different activities can be prepared on creating 3D modeling and the achievements of the activities can be examined.
- It can be suggested that more studies should be done on how to gain computational thinking and spatial thinking skills.

References

Akpınar, Y., Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *Elementary Education Online,* 13(1), 1–4.

Akyol Altun, C. (2018). *Algorithm and coding education in pre-school teaching program integration the effectiveness of problem solving skills in students*. Master Thesis, Ankara University, Türkiye.

Alyeşil Kabakçı, D., Demirkapı, A. (2016). The effect of "mathematics and art" course activity applications on students' spatial talents in Izmit science and art center. *HAYEF Journal of Education,* 13(1), 11–22.

Arslan Namlı, N., Aybek, B. (2022). An investigation of the effect of block-based programming and unplugged coding activities on fifth graders' computational thinking skills, self-efficacy and academic performance. *Contemporary Educational Technology,* 14(1), art. no. ep341. https://doi.org/10.30935/cedtech/11477

Atman Uslu, N., Mumcu, F., Eğin, F. (2018). The effect of visual programming activities on secondary school students' computational thinking skills. *Ege Journal of Educational Technologies,* 2(1), 19–31.

Atmatzidou, S., Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems,* 75, 661–670. http://doi.org/10.1016/j.robot.2015.10.008

Aydoğdu, Ş. (2020). The effect of block-based programming activities on pre-service teachers' computer programming self-efficacy and computational thinking skills. *Educational Technology Theory and Practice,* 10(1), 303–320. http://doi.org/10.17943/etku.649585

Bargury, I. Zur, Haberman, B., Cohen, A., Muller, O., Zohar, D., Levy, D., Hotoveli, R. (2012). Implementing a new computer science curriculum for middle school in Israel. *2012 Frontiers in Education Conference Proceedings*, 1–6. http://doi.org/10.1109/FIE.2012.6462365

Battelle for Kids. (2019). Framework for 21st century learning. Partnership for 21st Century Learning. Retrieved from http://www.battelleforkids.org/networks/p21/frameworks-resources

Baz, F.Ç. (2018). A comparative analysis of coding software for children. *Current Research in Education,* 4(1), 36–47.

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing,* 1(2), 67–69.

Carroll, J.B. (1993). *Human cognitive abilities: A survey of factor-analytic studies*. New York: Cambridge University Press.

Caeli, E.N., Bundsgaard, J. (2020). Computational thinking in compulsory education: A survey study on initiatives and conceptions. *Educational Technology Research and Development,* 68(1), 551–573. https://doi.org/10.1007/s11423-019-09694-z

Çavdar, L. (2018). *Evaluation of the online platforms used in teaching coding: Code.org*. Master Thesis, Tokat Gaziosmanpaşa University, Türkiye.

Çetin, S. (2019). *The effect of augmented reality applications on academic achievement, attitudes and spatial visualization skills of vocational school students in technical drawing course*. Master Thesis, Bursa Uludağ University, Türkiye.

Chaudhary, V., Agrawal, V., Sureka, A. (2016). An experimental study on the learning outcome of teaching elementary level children using Lego Mindstorms EV3 robotics education kit. In *Proceedings of the IEEE 8th International Conference on Technology for Education*, Bombay, India, 2–4 December 2016; pp. 38–41. https://doi.org/10.48550/arXiv.1610.09610

Clements, D.H. (1998). *Geometric and spatial thinking in young children* (Report no. NSF-MDR-8954664). Arlington, VA: NATIONAL Science Foundation. (ERIC No. ED436232).

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., *et al.* (2015). Computational thinking: A guide for teachers. Retrieved from Computing at Schools. https://community.computingatschool.org.uk/resources/2324/single

Demirkaya, C., Masal, M. (2017). The effect of geometric-mechanic games based activities on the spatial skills of secondary school students. *Sakarya University Journal of Education,* 7(3), 600–610. http://doi.org/10.19126/suje.340730

Denning, P.J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM,* 52(6), 28–30. http://doi.org/10.1145/1516046.1516054

Devon, R., Engel, R.S., Foster, R.J., Sathianathan, D., Turner, F.F.W. (1994). The effect of solid modeling on 3d visualization skills. *Engineering Design Graphics Journal,* 58(2), 4–11.

Dilmen, K., Kert, S.B., Uğraş, T. (2023). Children's coding experiences in a block-based coding environment: A usability study on code.org. *Education and Information Technologies*. `https://doi.org/10.1007/s10639-023-11625-8`

Engelhardt, K., Chioccariello, A., Ferrari, A. (2018). *Developing computational thinking in compulsory education: Implications for policy and practice*. Punie, Y., Kampylis, P. (Eds.) Publications Office.

Erol, O. (2015). *The effects of teaching programming with Scratch on pre-service information technology teachers' motivation and achievement*. Doctoral dissertation, Eskişehir Anadolu University, Türkiye.

Gagnier, K.M., Holochwost, S.J., Fisher, K.R. (2022). Spatial thinking in science, technology, engineering, and mathematics: Elementary teachers' beliefs, perceptions, and self□efficacy. *Journal of Research in Science Teaching,* 59(1), 95–126. `https://doi.org/10.1002/tea.21722`

Grover, S. (2018). The 5th 'C' of 21st century skills? Try computational thinking (Not coding) | EdSurge News. Retrieved from `https://www.edsurge.com/news/2018-02-25-the-5th-c-of-21st-century-skills-try-computational-thinking-not-coding`

Haladjian, J., Bredies, K., Brügge, B. (2016). Interactex: An integrated development environment for smart textiles. *International Symposium on Wearable Computers*, Digest of Papers, 12–16-Sept, 8–15.

ISTE (International Society for Technology in Education) (2015). Computational thinking. Retrieved from `http://www.iste.org/docs/ct-documents/ctleadershipt-toolkit.pdf?sfvrsn=4`

Jones, S.P., Liu, C.C., Cheng, Y.B., Huang, C.W., Kalelioglu, F., Bers, M.U., … Houlden, N. (2014). Computing at school in the UK: From guerrilla to gorilla. *Computers and Education,* 72(3), 1–13. `http://doi.org/10.1016/j.compedu.2013.10.020`

Kafai, Y.B., Burke, Q. (2013). Computer programming goes back to school. *The Phi Delta Kappan,* 95(1), 61–65. `https://doi.org/10.1177/003172171309500111`

Kalaycı, Ş. (2010). *SPSS Uygulamalı çok değişkenli istatistik teknikleri.* Ankara: Asil Publication,

Kalelioğlu, F., Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education,* 13(1), 33–50.

Kazakoff, E.R., Sullivan, A., Bers, M.U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal,* 41(4), 245–255. `http://doi.org/10.1007/s10643-012-0554-5`

Kelly, J.F. (2014). *3D modeling and printing with tinkercad*. Indianapolis.

Korkmaz, Ö., Çakır, R., Özden, M.Y. (2015). Computational thinking levels scale (CTLS) adaptation for secondary school level. *Gazi Journal of Engineering Sciences*, 1(2), 67–86.

Köse, E. (2010). *Bilimsel araştırma yöntemleri*. Ankara: Nobel Publication, 96–120.

Lye, S.Y., Koh, J.H.L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior,* 41, 51–61. `https://doi.org/10.1016/j.chb.2014.09.012`

Martín-Dorta, N., Saorín, J.L., Contero, M. (2008). Development of a fast remedial course to improve the spatial abilities of engineering students. *Journal of Engineering Education,* 97(4), 505–513. `http://doi.org/10.1002/J.2168-9830.2008.TB00996.X`

Mcclurg, P., Lee, J., Shavalier, M., Jacobsen, K. (1997). Exploring children's spatial visual thinking in an hypergami environment. VisionQuest: Journeys toward Visual Literacy. In *Selected Readings from the Annual Conference of the International Visual Literacy Association,* 257–266.

Mumcu, Y.H., Yıldız, S. (2015). Developing, implementing and evaluating of a web-based instructional material supporting spatial thinking. *Elementary Education Online,* 14*(*4), 1290–1306. `http://doi.org/10.17051/io.2015.48587`

Olkun, S. (2003). Making connections: improving spatial abilities with engineering drawing activities [Electronic Version]. *International Journal for Mathematics Teaching and Learning*. Retrieved from `http://www.cimt.plymouth.ac.uk/journal/default.htm`

Oluk, A., Korkmaz, Ö., Oluk, H.A. (2018). Effect of Scratch on 5th graders' algorithm development and computational thinking skills. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 9(1), 54–71. `http://doi.org/10.16949/turkbilmat.399588`

Özcan, K.V., Akbay, M., Karakuş T. (2016). Effects of gaming habits of university students on their spatial skills. *Kastamonu Education Journal,* 24(1), 37–52.

Pakman, N. (2018). *The effects of basic level coding, robotic, 3D design and game design education on the 8–10 years old students' problem solving and reflective thinking skills.* Master Thesis, Bahçeşehir University, Türkiye.

Portelance, D.J. (2015). *Code and tell: An exploration of peer ınterviews and computational thinking with ScratchJr in the early childhood classroom a thesis submitted by.* Doctoral dissertation, Tufts University, Medford, USA.

Rafi, A., Samsudin, K.A., Ismail, A. (2006). On improving spatial ability though computer-mediated engineering drawing instruction. *Educational Technology & Society,* 9(3), 149–159.
Šafhalter, A., Vukman, K.B., Glodez, S. (2016). The effect of 3D-modeling training on students' spatial reasoning relative to gender and grade. *Journal of Educational Computing Research,* 54(3), 395–406. https://doi.org/10.1177/0735633115620430
Sarı, D. (2012). *Researching the effect of the instruction of the transformational geometry assisted with the concrete models on the eighth grade students' attitudes towards geometry and spatial thinking.* Master Thesis, Dokuz Eylül University, Türkiye.
Scharf, F., Winkler, T., Herczeg, M. (2008). Tangicons: Algorithmic reasoning in a collaborative game for children in kindergarten and first class. Proceedings of the 7th *International Conference on Interaction Design and Children, IDC 2008*, 242–249. http://doi.org/10.1145/1463689.1463762
Scullard, S., Tsibolane, P., Garbutt, M. (2019). The role of scratch visual programming in the development of computational thinking of non is majors. In Pacis.
Seçer, İ. (2013). *SPSS ve LISREL ile pratik veri analizi: Analiz ve raporlaştırma*. Ankara: Anı Publication.
Senemoglu, N. (2018). *Development of learning and teaching Theory to Application [Gelişim öğrenme ve öğretim Kuramdan uygulamaya]*. Ankara: Pegem akademi.
Shin, S., Park, P., Bae, Y. (2013). The effects of an ınformation-technology gifted program on friendship using Scratch programming language and clutter. International *Journal of Computer and Communication Engineering,* 2(3), 246–249. https://doi.org/10.7763/IJCCE.2013.V2.181
Somuncu, B. (2021). *The effect of coding training on the mathematical reasoning skills of preschool children.* Master Thesis, Çukurova University, Türkiye.
Tağci, Ç. (2019). *Research of effects of coding education on primary school students.* Master Thesis, Afyon Kocatepe University, Türkiye.
Talan, T. (2020). Investigation of the studies on the use of scratch software in education. *Journal of Education and Future,* 18, 95–111. http://doi.org/10.30786/JEF.556701
Tedre, M., Denning, P.J. (2016, November). The long quest for computational thinking. 16th Koli Calling *International Conference on Computing Education Research*. New York, NY: ACM; 2016, pp. 120–129. https://doi.org/10.1145/2999541.2999542
Timur, S., Timur, B., Güvenç, E., Us, İ., Yalçinkaya-Önder, E. (2021). Pre-service pre-school teachers' opinions about using block-based coding/Scratch. *Acta Didactica Napocensia,* 14(2), 299–317, https://doi.org/10.24193/adn.14.2.22
Toma, R.B. (2023). Measuring acceptance of block-based coding environments. *Technology, Knowledge and Learning,* 28, 241–251. https://doi.org/10.1007/s10758-021-09562-x
Totan, H.N., Korucu, A.T. (2023). The effect of block based coding education on the students' attitudes about the secondary school students' computational learning skills and coding learning: Blocky sample. *Participatory Educational Research,* 10(1), 443–461. http://doi.org/10.17275/per.23.24.10.1
Uzunboylar, O. (2017). *Teachers and students views regarding coding education on secondary school level.* Master Thesis, Ege University, Türkiye.
Webb, H.C., Rosson, M.B. (2013). Using scaffolded examples to teach computational thinking concepts. SIGCSE 2013 – *Proceedings of the 44th ACM Technical Symposium on Computer Science Education,* 95–100. http://doi.org/10.1145/2445196.2445227
Weinberg, A.E. (2013). *Dissertation computational thinking: An investigation of the existing scholarship and research.* Doctoral dissertation, Colorado State University, Colorado, USA.
Weintrop, D., Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. *Proceedings of IDC 2015: The 14th International Conference on Interaction Design and Children,* 199–208. http://doi.org/10.1145/2771839.2771860
Wing, J.M. (2006). Computational thinking. *Communications of the ACM,* 49(3), 33–35
Yıldız, B. (2009). *The effects of using three-dimensional virtual environments and concrete manipulatives on spatial visualisation and mental rotation.* Master Thesis, Hacettepe University, Türkiye.
Yolcu, V. (2018). *The effect of using robotics on academic success, computational thinking skills and transfer of learning in programming education.* Master Thesis, Süleyman Demirel University, Türkiye.
Yurt, E. (2011). *The effects of modeling-based activities created via virtual environment and concrete manipulatives on spatial thinking and mental rotation abilities.* Master Thesis, Selçuk University, Türkiye.

M. Küçük is the founding partner and manager of Abranero Ltd. He received his Bachelor of Science degree in Department of Computer Education and Instructional Technology from Necmettin Erbakan University in 2018. Subsequently, he holds his Master's degree in the same discipline at Suleyman Demirel University, completing it in 2023. His research interests include computational thinking, spatial thinking and blog coding.

T. Talan is an Associate Professor in the Department of Computer Engineering at Gaziantep Islam Science and Technology University. He has received the BSc degree in Department of Computer Education and Instructional Technology from Inonu University in 2005; MSc degree in Educational Technology from Suleyman Demirel University in 2014 and PhD degree from Istanbul University Informatics Department in 2018. His research areas include meta-analysis, bibliometric analysis, blended learning, distance learning, augmented reality, and use of social media in education. He has published his academic works in national and international journals.

M. Demirbilek holds Doctoral degrees in Curriculum & Instruction and master's in Curriculum & Instruction specializing in Educational Technology from School of Education at the University of Florida. He also holds B.S. and M.S. degrees in electronics engineering from Istanbul University. Currently he works Educational Sciences Department in the Faculty of Education at Suleyman Demirel University. He worked as a Post-Doctoral Researcher at Games, Learning, and Society (GLS) research group at the University of Wisconsin-Madison (2008–2009) on mobile media project supported by The MacArthur Foundation. As a graduate faculty and visiting research scholar in Digital Worlds Institute at University of Florida (UF) between 2013 and 2014, he taught "Video Games & Startups" course and advised graduate students. Dr. Demirbilek joined several European Union founded projects as the partner and IT expert.