REPUBLIC OF TÜRKİYE

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

THE EFFECTS OF BLOCK-BASED PROGRAMMING ENVIRONMENTS ON MIDDLE SCHOOL STUDENTS' COMPUTATIONAL THINKING AND SCIENCE ACHIEVEMENT

Şükran SUNGUR

MASTER OF SCIENCE THESIS

Department of Mathematics and Science Education

Science Education Program

Supervisor

Assoc. Prof. Dr. Gülbin ÖZKAN

July, 2024

REPUBLIC OF TÜRKİYE

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

THE EFFECTS OF BLOCK-BASED PROGRAMMING ENVIRONMENTS ON MIDDLE SCHOOL STUDENTS' COMPUTATIONAL THINKING AND SCIENCE ACHIEVEMENT

A thesis submitted by Şükran SUNGUR in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE is approved by the committee on 08.07.2024 in Department of Mathematics and Science Education, Science Education Program.

Assoc. Prof. Dr. Gülbin ÖZKAN
Yıldız Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Gülbin ÖZKAN, Supervisor
Yildiz Technical University ______________________

Prof. Dr. Hakan AKÇAY, Member
Boğaziçi University ______________________

Prof. Dr. Bayram COŞTU, Member
Yildiz Technical University ______________________

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of "The Effects of Block-Based Programming Environments on Middle School Students' Computational Thinking and Science Achievement" supervised by my supervisor, Assoc. Prof. Dr. Gülbin ÖZKAN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Şükran SUNGUR

Dedicated to my family

ACKNOWLEDGEMENTS

Firstly, I would like to extend my deepest gratitude to my advisor, Assoc. Prof. Dr. Gülbin Özkan, whose support, invaluable guidance, and profound knowledge were instrumental throughout this study. I am equally grateful to my thesis committee members, Prof. Dr. Bayram Coştu and Prof. Dr. Hakan Akçay, for their insightful feedback, constructive suggestions, and significant contributions which greatly enhanced the quality of my study.

My gratitude also extends to Uludağ University professors and collogues supports me. Their academic support created an encouraging atmosphere that was vital to my research efforts. I sincerely appreciate all the assistance and encouragement that my graduate friends at Yıldız Technical University have given me.

Finally, I would like to express my deep appreciation to my father İlyas Sungur, my mother Leyli Sungur, and my siblings. Their moral support and belief in me have been a constant source of strength and inspiration. They have provided me with the resilience and motivation needed to overcome the numerous challenges that I encountered throughout my academic journey. Their encouragement and understanding have been instrumental throughout my life, guiding me to pursue my goals with determination and confidence. The love and support from my family have been invaluable, and I am eternally grateful for their presence in my life.

Şükran SUNGUR

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CT	Computational Thinking
CTt	Computational Thinking Test
FMKt	Force and Motion Knowledge Test
MoNE	Ministry of National Education
NGSS	Next Generation Science Standards

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Effects of Block-Based Programming Environments on Middle School Students' Computational Thinking and Science Achievement

Şükran SUNGUR

Department of Mathematics and Science Education

Science Education Program

Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Gülbin ÖZKAN

In this study, one of the block-based programming environments, Scratch, was integrated into education. The purpose of this study was to investigate whether providing science education to middle school sixth-grade students with a block-based programming environment would improve students' CT and science achievement. The study also aimed to investigate the opinions of students who receive science education through block-based programming about Scratch.

A quasi-experimental research design was used in the study. Participants were 51 sixth grade students in a public middle school selected according to the convenience sampling method. In the experimental group, teaching was carried out using Scratch programming environments; in the control group, teaching was carried out in accordance with the teaching method recommended by the Ministry of National Education. Three different data collection tools were used. These are computational thinking test, academic achievement test, and activity evaluation form. Computational thinking test and academic achievement test were applied to both groups as pre and post-test. In addition, the activity evaluation form was collected from the students in the experimental group after the intervention. In this study, the researcher administered the computational thinking test to 353 middle school students, carried out validity and reliability analyses, and adapted the test from English to Turkish.

The findings indicate that block-based programming environments are effective on students' computational thinking and science achievement. When the groups' computational thinking was analyzed, there was a significant difference in favor of the experimental group. Although there is no significant difference between the groups in the achievement test, the average score of the experimental group is higher than the control group. Additionally, most of the students expressed positive opinions about Scratch programming activities.

Keywords: Computational thinking, science education, Scratch, computational thinking test, force and motion.

Blok Tabanlı Programlama Ortamlarının Ortaokul Öğrencilerinin Bilgi İşlemsel Düşünme ve Fen Başarılarına Etkileri

Şükran SUNGUR

Matematik ve Fen Bilimleri Eğitimi Anabilim Dalı

Yüksek Lisan Tezi

Danışman: Doç. Dr. Gülbin ÖZKAN

Bu çalışmada blok tabanlı programlama ortamlarından biri olan Scratch eğitime entegre edilmiştir. Çalışmanın amacı ortaokul altıncı sınıf öğrencilerine blok-tabanlı programlama ortamı ile fen bilgisi eğitimi vermenin öğrencilerin bilgi işlemsel düşünmelerini ve fen başarılarını geliştirip geliştirmediği araştırmaktır. Çalışmada ayrıca blok-tabanlı programlama yoluyla fen eğitimi alan öğrencilerin Scratch hakkındaki görüşlerinin araştırılması da amaçlanmıştır.

Çalışmada yarı deneysel araştırma deseni kullanılmıştır. Katılımcılar, uygun örnekleme yöntemine göre seçilen bir devlet ortaokulundaki 51 altıncı sınıf öğrencisidir. Deney grubunda Scratch programlama ortamları kullanılarak, kontrol grubunda ise Milli Eğitim Bakanlığı'nın önerdiği öğretim yöntemine uygun olarak öğretim gerçekleştirilmiştir. Üç farklı veri toplama aracı kullanılmıştır. Bunlar, bilgi işlemsel düşünme testi, akademik başarı testi ve etkinlik görüş formudur. Bilgi işlemsel düşünme testi ve akademik başarı testi ön-test ve son-test olarak iki gruba uygulanmıştır. Ayrıca uygulama sonrasında deney grubundaki öğrencilerden etkinlik görüş formu toplanmıştır. Çalışmada kullanılan bilgi işlemsel düşünme testi araştırmacı tarafından 353 ortaokul öğrencisine uygulanmış, geçerlik ve güvenirlik çalışmaları yapılmış ve İngilizce'den Türkçe'ye uyanlanmıştır.

Çalışmanın sonuçlarına göre blok tabanlı programlama ortamları öğrencilerin bilgi işlemsel düşünmeleri ve fen başarıları üzerinde etkilidir. Grupların bilgi işlemsel düşünmelerine bakıldığında deney grubunun lehine anlamlı bir farklılık bulunmuştur. Başarı testinde gruplar arasında anlamlı bir fark çıkmamasına rağmen deney grubunun puan ortalamaları kontrol grubuna göre daha yüksektir. Ayrıca öğrencilerin çoğu, Scratch programlama etkinlikleri hakkında olumlu görüş belirtmişlerdir.

Anahtar Kelimeler: Bilgi işlemsel düşünme, fen eğitimi, Scratch, bilgi işlemsel düşünme testi, kuvvet ve hareket.

1
INTRODUCTION

With technology playing a significant and bigger role in both our daily lives and the classroom, it is obvious that students need to be equipped with the skills needed to think critically and use new technologies to suggest solutions to difficult problems. Although it is widely accepted that teaching students to solve problems and think critically is important, there does not seem to be a consensus on the best ways to teach these vital abilities in the classroom, including when and how to utilize instructional resources. However, studies have shown that (Yadav et al., 2014) one way to teach these abilities is to use computational thinking (CT), which has received attention for being a method to teach the fundamental 21st-century skills that each student requires for success in a world that is changing.

The process of problem-solving and decision-making that incorporates concepts and methods from computer science is known as CT (Wing, 2006, 2008). It entails dividing difficult problems into pieces, recognizing patterns and trends, creating algorithms, and applying abstraction to make solutions simpler and more universal (Wing, 2006). Many educators, researchers, and policymakers found significance in Jeannette Wing's article (Wing, 2008), which underlined the significance of CT in preparing children for the difficulties of the digital age. Moreover, CT aims to improve students' thinking skills on a qualitative level rather than just using digital technologies (Li et al., 2020; Moreno-León et al., 2018).

To meet the demands of 21st-century skills, students must be prepared with analytical, interpersonal, creative, and innovative abilities (Geng, 2021). But above all, they must be able to think critically to deal with the difficulties of this era. In addition, Nurhayati et al. (2020) stated that students need to develop the ability to think critically and learn efficiently to meet the demands of an evolving world. In accordance with the demands of the future world, CT is essential to equip students with the skills they require for success in the 21st century. CT improves students' capacity to solve complicated issues, facilitates creative thinking in a technologically advanced environment, and equips them to effectively adjust to new challenges in a variety of fields (Fraillon et al., 2019; Ogegbo & Ramnarain, 2022).

The increasing significance of CT in various aspects of life has enhanced the relevance of incorporating CT into educational practices (Güven & Gulbahar, 2020). Introducing CT into education is crucial (Moreno-León et al., 2018) as it provides students with vital skills required in the 21st century. Critical thinking, problem-solving, creativity, and analytical reasoning are some of the essential abilities. Studies have indicated that (Barr et al., 2011; Li et al., 2020) the integration of CT into the classroom enhances student achievement, especially in mathematics and science courses. In addition, students who engage in CT have a deeper understanding of the world and are better equipped for professions in a variety of topics beyond computer science and technology. Furthermore, incorporating CT into education provides more than just preparing students for future careers. It gives individuals the ability to take an active role in a world that is changing quickly, giving them the ability to adapt, produce, and make significant contributions to society (Higgins, 2014). Moreover, incorporating CT into the classroom is consistent with the Next Generation Science Standards (NGSS), which place a strong emphasis on the application of computational tools and techniques in scientific and engineering processes. According to the NGSS, students learn more about complex systems, human behavior, and scientific ideas when CT is incorporated into the classroom. Through this combination, students are given the tools to become active problem solvers and contributors of society as well as prepared for future professions.

The improvement of CT might be aided by a variety of interventions and technology, which can enhance academic performance as well as soft skills development. One way of incorporating CT into the classroom is by using block-based programming tools (Aksit & Wiebe, 2020). These tools are made especially to help beginners learn CT and programming concepts, which will facilitate their understanding and program-making abilities. MIT (Resnick & Siegel, 2007) created the well-known block-based programming tool Scratch. With Scratch, users may drag and drop code blocks to construct their own interactive stories, games, and animations. Therefore, students can interact with problems in a way that can be solved through CT and are actively involved in the creation and exploration process when using Scratch. Studies have demonstrated that these block-based programming environments can enhance students' achievement of key programming principles, including conditional, loops, and variables (Sáez-López et al., 2016). Furthermore, there is empirical evidence suggesting that block-based programming approaches can positively influence students' comprehension of scientific and mathematical concepts (Lai & Lai, 2012; Rodríguez-Martínez et al., 2020).

There are examples of how lessons incorporate CT in the literature from recent studies (Güven & Gulbahar, 2020; Ogegbo & Ramnarain, 2022). In this study, Scratch was used as a block-based programming tool to enhance students' CT within the CT framework developed by Brennan & Resnick (2012) and students' success in science topics. As Cetin (2016) stated that the incorporation of Scratch in educational settings improves students' understanding of programming concepts, encourages their enthusiasm for coding and computer science, and enhances their CT. In this context, integrating Scratch into science education not only fosters learning of programming skills but also acquires essential CT skills used in scientific inquiry (Lai & Lai, 2012). Incorporating Scratch, a block-based programming tool, into science instruction helps improve students' CT by introducing them to essential concepts like logic and sequence (Aksit, 2018) so this will enable students to solve problems and create interactive media in an organized and rational way.

The current study investigated integrating CT into science lessons for sixth grade students using a block-based programming environment. In a public middle school, a six-week classroom intervention was designed and then implemented to assist students understand concepts of CT as well as force and motion concepts using block-based programming tools. The current research adopts a quasi-experimental design, the force and motion unit was taught using Scratch in the experimental group, while the Ministry of National Education's recommended approach (supported by examples and activities from the Science 6 textbook) was followed in the control group. The Computational Thinking Test (CTt) (Román-González et al., 2017) and The Force and Motion Knowledge Test (FMKt) (Çimentepe, 2019) were applied as pre and post-test to the control and experimental groups. It was investigated whether the effects of the interventions caused a statistically significant difference in force and motion achievement and CT between the experimental and control groups. Additionally, the opinions of students in the experimental group regarding the integration of Scratch programming activities into science class were obtained qualitatively through an activity evaluation form.

1.1 Original Contributions

Research indicates that CT has not been adequately addressed in science classrooms or successfully incorporated into K–12 science curricula, even though it seems that everyone agrees that these practices and concepts are essential to education (Wilensky et al., 2014; Wilson, 2013; Yadav et al., 2016). This is a result of the lack of clear instructions that make it difficult to integrate CT into the classroom (Barr et al., 2011). In this context, this study contributed to the literature on (1) presenting empirical data on the usage of block-based programming tools, allows middle school students improve their CT and achievement, (2) adaptation of a valid and reliable computational thinking test (CTt) from English to Turkish (3) integrating block-based programming environment with science education to achieve interdisciplinary objectives, (4) bringing block-based programming tools into science class in order to improve students' CT and achievement to expand the literature on these applications, (5) sharing the opinions of students about Scratch who obtain science lessons integrated with block-based programming environment, (6) showing how activities prepared in Scratch can be integrated with science education. In this study, it has been observed that when science education integrated with one of the block-based programming programs, Scratch, students' CT and achievement improved. Also, this study demonstrated that students who get Scratch integrated science education have positive opinions about Scratch.

1.2 Significance of the Research

In classrooms, it is crucial that students interact with computer science because technological advancements have made them more widely available for use. Therefore, it should be possible for students to gain a fundamental comprehension of CT, which is necessary for contributing to the modern world (Angevine et al., 2017). In addition, CT promotes creativity and innovation, helps students become more critical thinkers and problem solvers, improves their academic achievement in disciplines like math and science, and gets them ready for professions in a technologically advanced society (Hurt et al., 2023; Li et al., 2020). For a variety of reasons, CT has not been effectively taught in schools or included into K–12 curricula. It's possible that many educators lack enough professional development or training in CT, or teachers frequently have the misconception that CT is just about using computers, which makes them hesitant to include it in the curriculum (Güven & Gulbahar, 2020). Additionally, it has been challenging to include CT into education due to a lack of definition and instruction on how to incorporate it into the curriculum (Lee et al., 2011). Using block-based programming tools (Aksit & Wiebe, 2020) is one way to integrate CT into the classroom. However, block-based learning environments are restricted to specific subjects (like computer science) and are typically continued as extracurricular activities (clubs and workshops, specialized course training, etc.) because they cannot be completely integrated with the curriculum. Thus, to facilitate the integration of education, this study will demonstrate to teachers how block-based learning environments affect students' CT and achievement in science teaching.

The significance of improving CT in children at a young age is becoming increasingly apparent, particularly considering the quickly evolving technological environment (Grover et al., 2015). Introducing CT to children at a young age can provide them with essential skills and a mindset needed to navigate the complexities of the digital environment and creatively solve problems (Voogt et al., 2015). In this regard, academic studies have been carried out to examine the most effective methods for integrating CT into education, such as involving specific educational approaches, instructional resources, coding languages, and curriculum classifications (Ogegbo & Ramnarain, 2022). In addition, CT is essential for getting the fundamentals of computer science and solving problems, which are useful in a variety of educational fields (Pietros et al., 2022). Consequently, there is a strong need to design curricula and learning environments by integrating block-based programming environments with science education to enhance students' CT and achievement. This study provides understanding on how instructional methods that use block-based programming environments, Scratch, might help students develop CT, which is regarded as necessary for success in the digital era and their achievement.

The current study investigated integrating block-based programming environments into science lessons for sixth-grade students to enhance students' achievement and

CT. In this study, the researcher administered the CTt (Roman-González et al., 2017) to middle school students, carried out validity and reliability analyses, and adapted the test from English to Turkish. The study employed a quasi-experimental research design. The control group received instruction in line with the Ministry of Education's (MoNE) suggested teaching methodology, whereas the experimental group used Scratch programming environments. In a public middle school, a six-week classroom intervention was designed and then implemented to assist students understand concepts of force and motion as well as CT concepts with block-based programming tools. The study also investigated the opinions of students who receive science education through block-based programming about Scratch. Recently, there has been a lot of interest in integrating block-based programming environments into science education as researchers and educators look for methods to improve students' achievement and CT (Lee et al., 2020; Weintrop, 2019). This study found that students' CT and success increased when science education was combined with Scratch, a block-based programming tool. Additionally, this study showed students who receive integrated science education using Scratch have positive opinions of Scratch. Therefore, the current study offers educators new techniques and materials for education by providing information about efficient teaching methods that foster the growth of CT and achievement of science topics.

1.3 The Purpose of the Study

The purpose of this study was to investigate whether providing science education to sixth grade students with a block-based programming environment would improve students' achievement of science concepts and CT. The study also aimed to investigate the opinions of students who receive science education through block-based programming environments about Scratch.

1.4 The Research Questions

I. Are there any significant differences in the computational thinking of sixth-grade students after the intervention?
II. Are there any significant differences in the computational thinking of sixth-grade students between the control and experimental groups?

III. Are there any significant differences in the achievement of force and motion unit of sixth-grade students between the control and experimental groups?

IV. Are there any significant differences in the achievement of the force and motion unit of sixth-grade students after the intervention?

V. What are the opinions of students studying science education integrated with block-based programming environments about Scratch?

1.5 Limitation and Assumption of the Study

The limitations and assumptions of the study should be considered when interpreting the findings. There are some limitations on this study.

- A particular sample of sixth-grade students from a public middle school with a computer laboratory participated in the study. Because the sample size is small, it may be challenging extending the results to a larger population.
- The experimental group in the current study receives an intervention that lasts for six weeks. After the intervention, differences within and between groups were examined using the CTt and FMKt post-test. The practice's long-term impacts are not examined in this study. Long-term consequences on students could be the subject of future research.
- It is assumed that the Scratch programming intervention was directly responsible for the observed gains in students' achievement and CT. The results might have been impacted by other external factors that were not included in the research.
- The research was carried out in a particular type of classroom. In other circumstances, variables including student demographics, school atmosphere, and instructor experience may have distinct effects on the results.
- The assessment tools used to measure CT and force and motion achievement may have limitations, so it is assumed that students provided honest and accurate responses on the assessment tool. Moreover, alternative or additional measures could enhance validity.

2
LITERATURE REVIEW

2.1 Computational Thinking

The idea of CT has been extensively debated in academics, but there is still disagreement regarding its precise definition. CT is an approach that makes use of fundamental computing concepts to understand human behavior and solve problems (Wing, 2008). According to Wing it is the process includes employing logical thinking, algorithmic reasoning, abstraction, and identifying patterns to examine issues and formulate resolutions. The concept of CT involves the capacity to engage in logical reasoning, problem-solving, and creating solutions using computational approaches (Selby & Woollard, 2013).

Being able to comprehend and make sense of processes and systems, both artificial and natural, CT uses ideas from computer science in a wide range of fields. For example, CT concepts can be introduced to students without computer programming. Peel et al. (2019) conducted one of the types of research in this field. In this research they investigated how introducing CT concepts to secondary school biology students can help them understand natural selection better. Students were taught CT concepts apart from computer programming (often known as "unplugged" CT) in a mixed methods research design. Students used these concepts to provide algorithmic explanations for the natural selection process. The key findings showed that following the intervention, students' understanding of natural selection increased, misconceptions decreased, and they were able to successfully explain natural selection using CT principles. The findings also demonstrated that teaching natural selection to students using CT not only improved their knowledge of the subject matter but gave them a transferable understanding that they could use in a variety of biological contexts. Another study conducted by Aytekin & Topçu (2023) designed plugged and unplugged courses that incorporated CT elements into the circulatory system topic. A quasi-experimental design was used to examine the effect of these instructional modules on sixth-grade students' CT abilities and achievement. The study's findings demonstrated that both experimental groups with plugged and unplugged courses had noticeably greater achievement in CT abilities and achievement of the circulatory system compared to control group.

Studies have indicated that CT is crucial during the preschool years as well. One of these studies was conducted by Kotsopoulos et al. (2022). To identify instances of CT in children's casual play, researchers examine images and recordings of kids in school. The researchers and teachers used a range of frameworks to examine the children's interactions and behaviors to search for signs of CT. Additionally, the researchers observe images and videos of children in educational settings. The study's outcomes emphasize how play-based and unplanned activities can involve CT even in the absence of technology, demonstrating the potential for early childhood educators to support young learners in acquiring CT. The study demonstrates that it is possible to integrate CT into unplugged activities in early childhood education settings. Another study on the application of CT in early primary education is conducted by Boticki et al. (2018). Students in the first grade of a primary school were utilizing a specially created CT tool in their classes. To determine how successfully students perform such tasks, the program made it possible to build and distribute digital assignments using CT content from three topic areas. The program made it possible for automatic task solution evaluation, including animations and visualizations that showed the precise steps that the students had selected and gave them the option to change their selection if necessary. Results indicate that early primary kids' performance on CT tasks across school topics is influenced by their past reading and math skills.

Studies also have been done on how to use programming tools to introduce CT into the classroom. An early childhood STEM magnet school conducted an intense robotics workshop for a week, organized by Kazakoff et al. (2013). 27 students participated in a robotics and programming class that lasted one week. Children in the workshop programmed robot behaviors using the tangible programming language CHERP. A picture-story sequencing task was used in the study to assess the participants' sequencing skills, one of the CT concepts, both before and after the intervention. In comparison to their pre-test results, students who took part in the one-week robotics and programming session had a substantial improvement in their post-test sequencing scores. This demonstrates that in comparison to a control group, the children's sequencing skills increased greatly after the workshop.

2.2 Computational Thinking in Education

CT can enhance students' problem-solving skills and critical thinking as well as their general digital technology competency when it is incorporated into the classroom environment (Güven & Gulbahar, 2020). Therefore, according to Voogt et al. (2015), CT is a "universal competence" that needs to be a part of each child's cognitive toolset. Nevertheless, there have been difficulties with integrating CT into educational curricula, with an ongoing gap between the acceptance of its value and its general implementation in educational institutions. The lack of a clear and unified understanding of CT's components and the best ways to teach and evaluate it is one of the main causes of this gap (Ogegbo & Ramnarain, 2022; Yadav et al., 2014, 2016). Consequently, the development of curricula, materials, and professional development programs to facilitate the incorporation of CT into the classroom has been the focus of numerous educational organizations and efforts (Aksit, 2018; Grover et al., 2015).

To find out how CT modules affect preservice teachers' attitudes toward computing and their comprehension of CT Yadav et al. (2014) organized quasi-experimental research. A CT module was given to the treatment group while it was not given to the control group. The findings showed that the CT program had an impact on preservice teachers' comprehension of CT concepts and their ideas about incorporating CT into their own classrooms. Preservice teachers who completed the module specifically perceived CT as a method of problem-solving that generates general principles through abstraction and algorithmic reasoning. Another research conducted with pre-service teachers by Mouza et al. (2017) designed an educational technology course aimed at teaching pre-service teachers how to incorporate CT into K–8 instruction. Then investigated how the course affects the attitudes and understanding of CT principles among pre-service teachers, as well as how to integrate this knowledge with content and methodology to support effective learning outcomes for students. The instruction had a positive impact on pre-service teachers' comprehension of CT ideas, methods, and resources, according to the results.

A study carried out in the fields of mathematics for grades K–12, which aims to determine whether teaching students to use coding in the classroom might improve their learning outcomes, is presented. For this reason, Calao et al. (2015) have planned a quasi-experimental study. The experimental group, which received training in Scratch, exhibited a statistically significant gain in their comprehension of mathematical processes, according to the results. Additionally, according to another study's findings conducted by Tsarava et al. (2022) CT was positively correlated with verbal reasoning, nonverbal visuospatial abilities, and complicated numerical abilities. In addition, Hsu & Liang (2021) conducted in the domains of English to help students enhance their multidisciplinary competencies in both English and CT, carried out both plugged and unplugged approaches. In the plugged strategy, educational robots were adopted to improve CT and offer practice with English vocabulary and sentences through a board game. While it was intended for studying conversational sentences and some foreign language vocabulary, the unplugged version of the educational board game featured playing a standard board game without a computer. The findings demonstrate that both strategies were beneficial for raising students' CT competency and English proficiency in target vocabulary and phrases.

Çiftçi & Topçu (2023) conducted experimental research with pre-service teachers about CT instruction in STEM courses that incorporate CT. Looking at the post-test scores on the scale of self-efficacy beliefs towards CT, the study's findings showed a substantial difference in favor of the experimental group, which received STEM education that included CT. The improvement of the pre-service teachers' attitudes about their ability to teach CT was found to be enhanced by the incorporation of CT within STEM education. One of the other studies carried out with CT and STEM was done by Hutchins et al. (2020). To develop, assess, and improve the Collaborative, Computational STEM (C2STEM) learning environment, they used a design-based methodology to learn about physics and CT principles. Students in this study constructed models that explain how objects move. Hutchins et al. (2020) examined student learning over the course of a semester-long study done in a high school physics classroom. It has been observed that even when students ran into problems, they generally appeared to be more encouraged and worked deeper on their physics assignments. Due to the studies carried out in the field of CT and STEM Basu et al. (2013) have created CTSiM, visual-programming, scaffolded, a cross-domain, and agent-based learning environment for middle school science, by taking use of the synergies between STEM fields and CT. They introduce and explain the CTSiM architecture and its methodology. The research is a case study outlining the difficulties that a student with high grades and low grades had when using CTSiM for ecological and kinematics lessons. Their study demonstrated the efficacy of CTSiM in enhancing learning outcomes for both science areas using a kinematics and ecology unit.

One of the ways to integrate CT into education is Scratch. Scratch was used in the mathematics class at the secondary school level (Rodríguez-Martínez et al., 2020). This experimental investigation suggests that Scratch can be used as an efficient teaching tool to assist students in understanding CT and mathematical concepts. Based on assessments conducted before and after the teaching, the sixth-grade students who engaged in the study showed increases in their comprehension of mathematical topics and their development of CT. According to the study, studying with Scratch can improve students' understanding of mathematical concepts and enhance their CT. Another study conducted by Marcum-Dietrich et al. (2024) created a four-week middle school curriculum unit that involved research, development, and implementation over the course of a five-year project. The project was an innovative computational weather forecasting attempt that integrated students' knowledge of meteorology with CT concepts and practices. To improve students' scientific understanding and their proficiency with CT abilities and procedures, the project resulted in the development of a deliberate instructional sequence incorporating disciplinary science and CT practices. Students' scientific understanding of CT practices and content considerably improved because of having the opportunity to use their CT abilities in class discussions. Therefore, teaching at all levels, including middle school, preschool, and elementary school, can benefit from the integration of CT as a desirable ability. Students can acquire higher-order thinking skills and learn how to use digital tools efficiently by integrating CT into classes (Ogegbo & Ramnarain, 2022).

2.3 Block-Based Programming Environment in Education

Block-based programming environments are becoming more and more common in classrooms. Instead of entering code, students may construct and interact with visual programming blocks in environments like Blockly, Scratch, and MIT App Inventor, which enhance the accessibility and engagement of the programming process for learners (Dutta & Mathur, 2011). When studying block-based programming, which demonstrates code using drag-and-drop blocks, offer a more engaging and readily available way than traditional text-based approaches (Weintrop, 2019). Scratch is an online visual block-based programming language developed by MIT Media Lab (Resnick & Siegel, 2007). Students can create projects online and make them into anything using fundamental block coding in Scratch. In addition, students can utilize their problem-solving abilities to produce interactive media, including animations, games, stories, and simulations, by using Scratch (Sáez-López et al., 2016). Through these activities, students not only enhance their CT skills but also get a deeper understanding of scientific subjects as they use these skills in real-world situations (Cetin, 2016).

Aksit & Wiebe (2020) examined the effects of teaching simulation-based modeling to middle school students using CT and block-based programming on their comprehension of CT abilities and force and motion principles. The study investigated how seventh-grade students' achievement of force and motion concepts and CT skills were affected by a week-long intervention course that introduced simulation-based model building and CT practices through a block-based programming environment in middle school science classrooms. The study included qualitative data from student interviews, classroom observations, and reflections from 82 seventh-grade students in addition to quantitative data from pre and post-tests. The findings showed that students' CT and achievement of force and motion concepts had significantly improved. Another study created with Scratch by Sáez-López et al. (2016) evaluated how a visual programming language is used in the classroom by examining the attitudes and results of primary school children in the fifth and sixth grade from five different schools in Spain. The goal of the project is to assess the efficacy of visual blocks programming and coding integration in the setting of elementary education in science and art lessons throughout a two-year academic period. According to the study, elementary school students who used Scratch made considerable progress in their understanding of computational practices, logic, and programming ideas. Along with increases in CT and computational practices, students showed greater motivation, enjoyment, passion, and commitment. The findings highlight the value and efficacy of employing active learning strategies like project-based learning to integrate visual programming languages like Scratch in elementary education. Another study designed by Tucker-Raymond et al. 2019) computer games addressing climate change with a curriculum that was based on the triadic game design framework, emphasizes play, reality, and meaning for eighth-grade students. Using field notes, screen capture software, video, and audio recordings, teachers watched and recorded student interactions during both planned critique sessions and unplanned, student-initiated critique sessions. According to the study, middle school students engaged more in CT skills when creating games, particularly in student-driven critique sessions where they examined user experience, problem-solving techniques, modeling, and elegant solutions. This suggests that although the unplanned criticisms focused more on the gameplay aspect, they nevertheless gave students a chance to interact with and consider significant aspects of CT. Additionally, by reviewing games, students were able to interact with fundamental ideas in CT and use those ideas to iteratively enhance their own designs.

According to Ardito et al. (2014) students who took part in a robotics intervention and those who did not did not show statistically significant differences in their state mathematics test scores. However, they did discover that a greater number of children who took part in the robotics intervention outperformed their classmates in the subjects of measurement, statistics, and algebra. Additionally, the teacher noticed that students participating in this study had improved in both their teamwork and problem-solving skills. In another study, (Nugent et al., 2016) middle school students who took part in robotics competitions, clubs, or camps showed improvements in programming, engineering design, and achievement in addition to higher levels of self-efficacy compared to a control group. In the setting of robotics summer competitions, clubs, or camps Nugent et al. (2016) observed greater success in raising kids' STEM self-efficacy. Another study studied how the Dash robot and Blockly software may be used to incorporate CT into elementary science instruction over a four-week summer camp program. Important CT ideas like loops, sequences, and conditionals were included into a lesson on the reproductive cycle of flowerless plants in this program. Based on participant drawings and Blockly code analysis, their findings revealed significant improvements in the children's CT techniques. Additionally, the participants' effective participation highlights the possibility that science units with CT integration could engage students and promote cognitive learning outcomes.

Erümit (2020) designed three distinct kinds of Scratch exercises to study 423 sixth-grade students' attitudes toward algorithmic thinking skills, computer technologies and reflective thinking skills in problem-solving. Results show that students' views toward computer technology were improved by animation activities. The problem-solving abilities of reflective thinking and algorithmic thinking were enhanced by the mathematical and game-planning exercises. Another study (Cetin, 2016) used a mixed-method design that combined quantitative and qualitative approaches to discuss preservice teachers' learning and attitudes toward programming to assess the effects of Scratch-based instruction. The findings showed that preservice teachers who were taught using Scratch had a considerably greater comprehension of computational concepts than those who were not. Additionally, they were more enthusiastic about programming. The findings indicated that preservice teachers' programming skills were improved and their attitudes about programming were positively influenced by Scratch-based instruction.

Grover et al. (2015) created a seven-week Scratch-based CT course for seventh and eighth students to develop algorithmic thinking through Scratch. The findings demonstrate that the progress in programming skills from the pre-test to the post-test was used to measure CT gain. This quasi-experimental research investigated whether teaching methods, face-to-face versus face-to-face + online, supported deep learning regarding computational concepts including decomposition, algorithms, loops, and conditionals. The results showed that students in the face-to-face + online group outperformed those in the face-to-face group, and both methods result in much higher CT improvements. Additionally, both groups were able to effectively apply their programming knowledge and abilities to problems using text. Another study (Oluk et al., 2018) investigates how the use of Scratch can help with algorithm development and enhance CT abilities. The study, a quasi-experimental research task, was created during six weeks in the fifth-grade software and computer technology course. In this process, the students in the experimental group were taught the algorithm using Scratch, whereas the students in the control group were taught the algorithm using the current curriculum. The study's findings showed that the experimental group's students' abilities in algorithm development and CT improved noticeably more than those of the control group. Both studies (Grover et al., 2015; Oluk et al., 2018) reveal that Scratch is a learning tool that can be used to improve CT skills.

2.4 Using Block-Based Learning Environments for Achievement

Using block-based programming environments in the classroom is one strategy to improve students' achievement. The potential advantages of these block-based programming environments for improving students' achievement have been investigated in recent studies. Enhancing students' achievement of scientific phenomena is one of the main advantages of implementing block-based programming in science classes. Students can experience and explore complex scientific subjects in a more practical and engaging way by developing interactive models and simulation (Weintrop, 2019) so, deeper understanding and better learning outcomes could arise from this programming environment.

In this respect, studies have shown that programming environments based on blocks can have a favorable effect on students' achievement of scientific ideas. For instance, a study by Adler & Kim (2018) revealed that learners who were introduced to programming through a block-based platform exhibited a superior understanding of specific scientific concepts, such as Newton's laws of motion. Another study (Lai & Lai, 2012) examined how well Taipei fifth grade students learned science with the use of Scratch programming. This study employed a single group in a quasi-experimental design. For enhancing students’ knowledge with computer science, 96 fifth graders at an elementary school engaged in 15 weeks of Scratch programming programs. According to the study's findings, students preferred creating their own Scratch programming projects during their science classes and said they would be open to doing more of these kinds of computer programming activities for future science units.

Encouraging students to use block-based programming environments like Scratch in science classes can have a big impact on their scientific literacy. A critical educational goal is scientific literacy, which is described as the knowledge and comprehension of scientific ideas and processes necessary for individual decision-making, involvement in social and cultural issues, and economic growth (Roberts, 2013). Additionally, children that participate in science education using block-based programming are exposed to multidisciplinary learning, which combines scientific inquiry with computational thinking. This multidisciplinary strategy is in line with the goals stated by groups like the NGSS Lead States (2013), which supports integrated learning as a means of developing a flexible and adaptable scientific literacy. Students practice developing hypotheses, testing theories, evaluating evidence, and refining answers through programming exercises such as those offered by Scratch. These are fundamental scientific processes described in the NGSS Lead States (2013).

Researchers investigated how block-based programming environments might improve students' learning and understanding in a variety of subject areas. For example, Tabach et al. (2008) conducted research with seventh-grade students in a computer-intensive setting throughout a one-year basic algebra course. The results showed that block-based platforms empower students to freely experiment with and explore mathematical concepts through programming, which can help them build a more sophisticated knowledge of equation solving, grammatical principles, and symbolic abstractions. Furthermore, students' understanding of how to interpret and relate mathematical ideas to real-world situations may be improved using graphical representations and virtual instruments made possible by block-based programming (Minda et al., 2015). The data points to the possibility of greatly improving students' achievement of mathematics using block-based programming environments (Goldenberg & Carter, 2021; Kajander, 2018; Minda et al., 2015; Tabach et al., 2008; Zhong & Xia, 2020). Additionally, some studies show that using block-based programming environments in the classroom not only assist students by improving their achievement of programming, but also help students build cognitive abilities that are directly related to learning a language (Sands, 2019; Scherer, 2016; Yusoff et al., 2020). Also, according to the results of a study conducted in social studies (Luxton-Reilly et al., 2018) students can design a game that examines the difficulties faced by various parties in a dispute resolution scenario, or they might develop a simulation that simulates how government policies affect the distribution of wealth within a nation. In conclusion, through the process of creating and implementing these interactive projects, students get an in-depth understanding of the fundamental concepts and principles that influence political, social, and economic structures.

2.5 Theoretical Framework

It has become clear that introducing CT is a vital ability that should be included in the curriculum. This ability includes methods for solving problems, logical reasoning, and algorithmic thinking that are applicable in a variety of contexts and fields. Seymour Papert first proposed the idea of CT in 1980, According to Papert, CT involves using computers as a tool to encourage creative thinking and provide students the opportunity to study in a more engaged and adventurous way (Papert, 1980). According to Papert, the concept of "computational thinking" refers to assisting students in expressing themselves via the "language" of computers, thus encouraging a kind of problem-solving that makes use of computer science ideas like abstraction, coding, and algorithms.

Papert's constructionist educational theory is the basis of his perspective on CT. Constructionism is derived from Piaget's theory of constructivism, which assumes that people create their own knowledge and connect experiential learning to active engagement (Chuechote et al., 2020). Piaget's theory of constructivism supposes that meaningful experiences and prior knowledge influence how people respond to new situations and environments, which helps people develop achievement (Fosnot & Perry, 1996). Constructionism theory states that (Halverson & Sheridan, 2014) when students produce physical items, schemas are progressively built. Papert (1980) thinks that for students to properly understand programming and computational concepts, they need to be actively involved in generating artifacts, and that social and affective involvement is critical to the learning process. His theory is that students might have a deeper and personal understanding of the subject by creating something. According to Papert's (1993) theory of constructionism, children will do better if they create the specialized knowledge they need to complete tasks on their own. Additionally, when children actively participate, they will cognitively construct knowledge based on the information that they learn from activities and teachers (Papavlasopoulou et al., 2019; Papert, 1993).

Jeanette Wing popularized the term in 2006, leading to its wider use (Wing, 2006), He argues that CT goes beyond programming and encompasses system design, problem solving, and understanding human behavior by utilizing computer science principles. Wing states that (Wing, 2008) CT is a mindset and set of skills that can be applied universally, and which individuals from various fields would be motivated to acquire and apply, not just those in the computer science field. CT is emphasized by Wing's method as a necessary ability not just for computer scientists, but also for everyone. It views CT as a collection of techniques and strategies for solving problems that come from the study of computer science and can be widely used in all areas. Wing promotes the incorporation of CT in school curricula as a means of giving students the tools they need to comprehend and create in the digital world.

While CT has gained a lot of attention in recent years, there is not enough agreement on what CT involves (Barr et al., 2011; Selby & Woollard, 2013). The definition of CT that Brennan & Resnick (2012) developed while working with Scratch includes three essential dimensions which are computational perspectives, computational practices, and computational concepts. To explain the perspective shifts observed in young individuals working with Scratch, Brennan & Resnick (2012) introduced the dimension of perspectives designers create about themselves and the world around them. The researchers propose that these CT perspectives are questioning, connecting, and expressing. Brennan & Resnick (2012) framework consisted of CT practices they saw children participating in while working on their projects. Instead of focusing on "what you are learning," the authors propose that these computational exercises focus on "how you are learning." Four main categories of practices were identified by Brennan & Resnick (2012) as being iterative and incremental, abstracting and modularizing, remixing and reusing, testing and debugging. Seven computational concepts were found by Brennan & Resnick (2012) to be extremely helpful when using Scratch programming language in their research. They also propose that these concepts are applicable in non-programming and other programming contexts. Sequences, parallelism, loops, operators, conditionals, data, and events are all included in these concepts.

CT frameworks associated with Scratch (Brennan & Resnick, 2012) Agent Sheets (Repenning et al., 2000) and Weintrop et al. (2016). In this study, the block-based programming tool, namely Scratch, was incorporated into science class to improve students' CT by incorporating the CT concepts of the Brennan & Resnick (2012) framework. This study provides an understanding on how instructional methods that use a block-based programming environment, Scratch, might help students develop their CT. CTt was used to examine students' CT. Definitions of the concepts of CT, which are the subfactors of CTt, namely, sequences, parallelism, loops, operators, conditionals, and events are given below.

2.6 Definition of Terms

This section consists of definitions of the key terms associated with this study.

2.6.1 Computational Thinking Concepts

CT concepts are the basic building blocks of computer programming (Brennan & Resnick, 2012), which are employed in data manipulation and problem-solving. These concepts include loops, sequences, events, parallelism, operators, conditionals, and data. These concepts are crucial (Bian et al., 2022) for creating and executing algorithms and programs that are both effective and efficient. Each of these CT concepts is defined in the following sections.

2.6.1.1 Sequences

In computer programming, sequences (Brennan & Resnick, 2012) define the precise order in which actions or instructions are carried out. This order is essential for managing a program's flow and ensuring that tasks are completed in an appropriate order. For example, the pen on the screen can be programmed to draw triangles following the sequence of instructions. A set of instructions for drawing a triangle in a Scratch is shown in Figure 2.1. The steps are given below.

1. Move the pen down
2. Move 120 steps
3. Turn left for 120 degrees
4. Move 120 steps
5. Turn left for 120 degrees
6. Move 120 steps
7. Turn left for 120 degrees

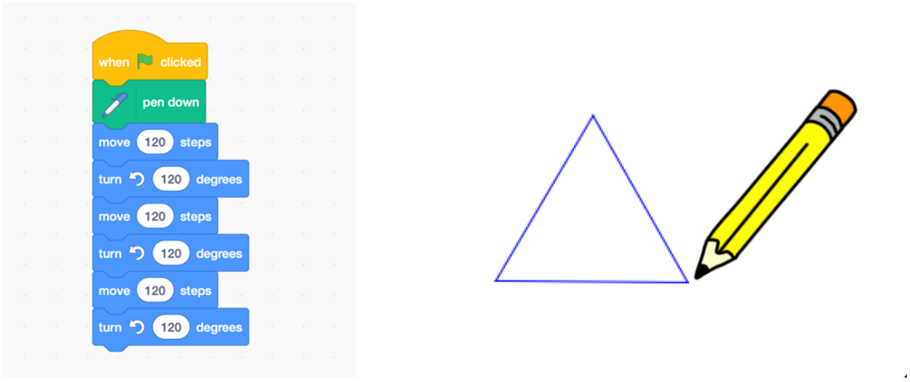


Figure 2.1 Example of a sequence of orders that create a triangle in Scratch

2.6.1.2 Loops

Loops (Brennan & Resnick, 2012) are necessary to repeat a given set of instructions until the specified requirement is met. They offer an efficient way to manage frequently running programs. Loops are represented by "repeat" blocks in Scratch which is a block-based programming language. In the previous example, to create a triangle, the pen had to move 120 steps and turn left for 120 degrees, and this was repeated three times. In total, rather than giving all these code blocks one by one, it can be stated how many times these blocks were repeated. Figure 2.2 shows how a loop can be used clearly and concisely to draw a triangle.

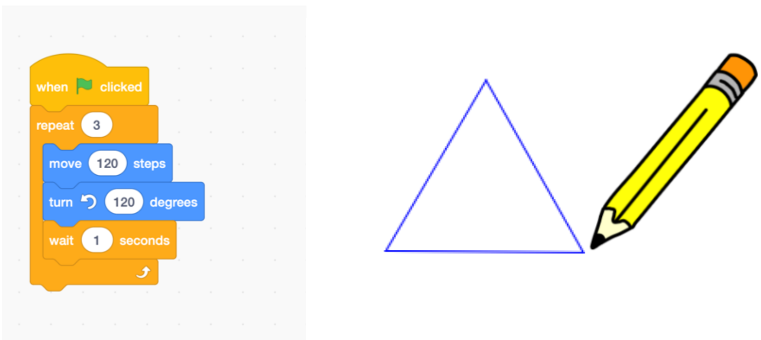


Figure 2.2 Example of a loop that creates a triangle in Scratch.

2.6.1.3 Parallelism

Parallelism is the capacity to carry out several tasks at the same time. It enhances efficiency and speed by using the capabilities of multi-core processors. This makes it possible to significantly increase productivity and performance (Zendler &

Spannagel, 2008). Parallelism (Brennan & Resnick, 2012) across objects is supported by Scratch. Parallelism within a single object is also supported by Scratch. In Figure 2.3 the sprite is designed to respond to the when green flag clicked event by carrying out three sets of actions simultaneously: (1) set to; (2) ask and wait; and (3) continuously repeat.

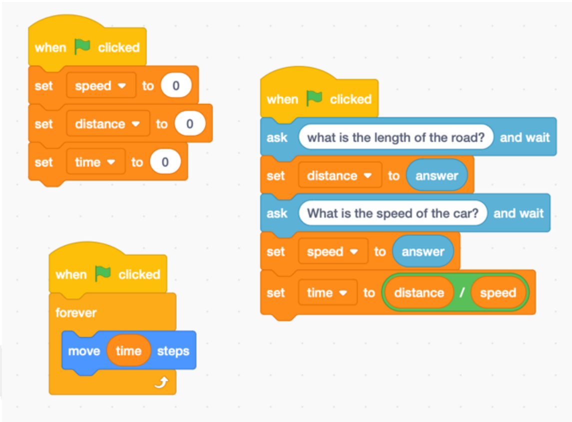


Figure 2.3 Example of parallelism within a single object in Scratch

2.6.1.4 Events

In a programming environment, events (Brennan & Resnick, 2012) are triggers or occurrences that start a reaction or an action. These events may be the result of system notifications, user activities, or the occurrence of certain circumstances. An event will result in an action in several conditions, as shown in Figure 2.4: (1) the object will turn forever in 15-degree increments when the green flag is clicked, and it follows the ongoing sequences. (2) when the space key is pressed, the object will change its position y by -10 and it follows the ongoing sequences. (3) when this sprite clicks, the object will go to x:0, y:0 and it follows the ongoing sequences. (4) when I receive a message, say hello for 2 seconds.

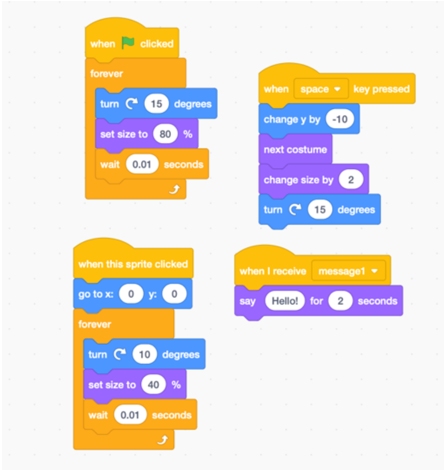


Figure 2.4 Example of code blocks representing types of events in Scratch

2.6.1.5 Conditionals

Statements known as conditionals (Brennan & Resnick, 2012) give the program the ability to decide depending on specific circumstances. They are necessary for regulating a program's flow and running code blocks in response to different circumstances. Scratch that makes use of the "if" block introduces this concept. In other words, a specific sequence of instructions is followed if a given condition is met. If not, an alternative set of instructions might be provided. In the example below, you could find the if conditional used in the balloon-popping game in Figure 2.5.

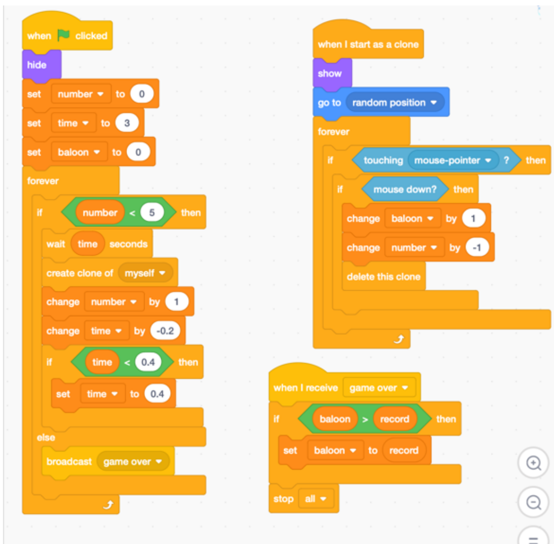


Figure 2.5 Example of conditionals in Scratch determining the behavior of the Sprite

2.6.1.6 Operators

Operators are keywords or symbols that manipulate data (Brennan & Resnick, 2012). They can be applied to several tasks, such as logical comparisons and mathematical computations to alter the data as needed. Numerous mathematical operations, such as subtraction, addition, division, and multiplication, as well as functions like sine and exponents, are supported by Scratch. It also supports string operations. The operator blocks used in Scratch are shown in Figure 2.6.

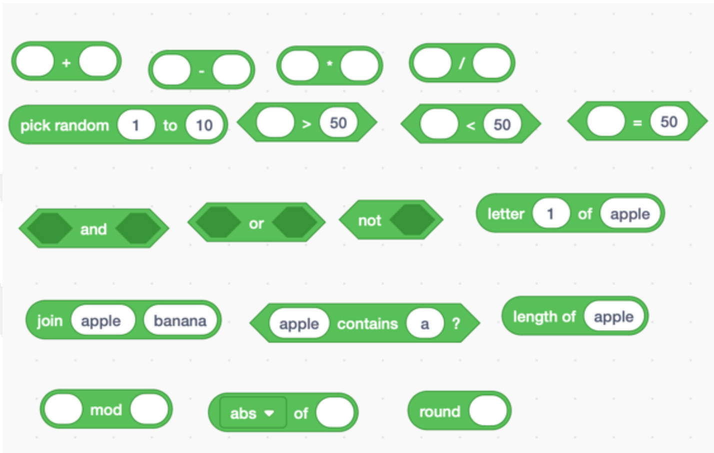


Figure 2.6 The operator blocks used in Scratch

3
METHODOLOGY

This chapter provides details regarding the study's design, the study's sample, the instruments used for data collection, the treatment administered, and the analysis of the gathered data.

3.1 Design of the Study

The current research adopts a quasi-experimental design, which is utilized when an experimental design is impractical. This design is commonly employed in educational settings where researchers may need to utilize existing groups. While quasi-experimental design involves assignments to groups, they lack random assignment of participants to these groups (Creswell-John & Creswell-David, 2017). The effectiveness of the intervention given to the experimental group was assessed in this study using a quasi-experimental approach with a pretest-posttest control group design. Two different groups were chosen for the study, one to be used as the experimental group and the other as the control group. Through conducting pretest and posttest assessments, it enabled a comprehensive examination of the intervention's effect on the experimental group in comparison to the control group.

Table 3.1 Phases of the Experiment

Group	Pre-test	Lesson	Post-test	Activity Evaluation Form
Experimental Group	CTt and FMKt	Lesson integrated with Scratch	CTt and FMKt	Activity Evaluation Form after implementation
Control Group	CTt and FMKt	Lesson without Scratch	CTt and FMKt	

Before the implementation, the researcher provided 2-week Scratch training to the science teacher at the school where the research was conducted. Data were obtained from both groups as a pre-test prior to implementation on the FMKt and CTt. Subsequently, the force and motion unit was taught using block-based programming environments through Scratch activities in the experimental group designed to enhance students' CT and achievement, while the control group received instruction using the Ministry of National Education's recommended teaching method (utilizing examples and activities from the Science 6 textbook).

At the end of the four-week intervention period, the FMKt and CTt were administered to both groups as a post-test. Additionally, the opinions of students in the experimental group regarding the integration of Scratch programming activities into science class were obtained through an activity evaluation form. Throughout these processes, the researcher remained in an observer position.

Two dependent variables are in the study CTt scores and FMKt scores. The methods of teaching force and motion topics are block-based programming environments, and the teaching method recommended by MoNE are independent variables of the study.

3.2 Sample of the Study

The participants in this study were two separate sixth-grade classes studying at a public middle school in Istanbul. The sample selection process was impacted by the research's goal. This school was selected because it has a computer laboratory, and all computers are functioning smoothly, with each student having access to a computer. Therefore, to identify the participants, purposive sampling was employed because it is based on the goal of the study (Fraenkel-Jack et al., 1993).

Two classrooms, one for experimentation and the other for control, were chosen at random. Groups defined as control and experimental were chosen at random. 23 sixth-grade students ranging in age from 11 to 13, were placed into the experimental and 28 sixth-grade students ranging in age from 11 to 13 were placed into the control groups. Gender equality was not taken into consideration because a comparison by gender was not included in the study problems.

Table 3.2 Distribution of the groups.

Grade Level	Experimental group	Control group
6	23	28

3.3 Data Collection Instruments

The instruments used to collect data for the study are described in this section. Data was collected using three different tools: The Computational Thinking Test (CTt) (Román-González et al., 2017), The Force and Motion Knowledge Test (FMKt) (Çimentepe, 2019) and the activity evaluation form.

The data analysis approach, data collection time, and data collection instruments for each sub-problem are presented in Table 3.3.

Table 3.3 Data collection instruments, data collection time, and data analysis

Sub-Problems	Data Collection Instruments	Data Collection Time	Data Analysis Approach
Are there any significant differences in sixth-grade students' computational thinking after the intervention?	CTt	pre-test and post-test	Paired sample t-test
Are there any significant differences in the sixth-grade students' computational thinking between the control and experimental groups?	CTt	post-test and post-test	Independent sample t-test

Table 3.3 Data collection instruments, data collection time, and data analysis (continued…)

Are there any significant differences in sixth-grade students' understanding of the force and motion unit after the intervention?	FMKt	pre-test and post-test	Paired sample t-test
Are there any significant differences in the sixth-grade students' understanding of force and motion unit between the control and experimental groups?	FMKt	post-test and post-test	Independent sample t-test
What are the opinions of students studying science education integrated with block-based programming environment about Scratch?	Activity Evaluation Form	at the end of the research	Content analysis

3.3.1 Computational Thinking Test (CTt)

The CTt was developed by Román-González, Pérez-González and Jiménez-Fernández (Roman-Gonzales, 2015; Roman-González et al., 2017). The purpose of the test is to evaluate the level of CT among students. Originally, the CTt was designed as a 40-item test (Roman-Gonzales, 2015). Following that, CTt is given to 1,251 Spanish students ranging in grade levels from fifth to tenth. The test's validity and reliability were examined, and its item count was lowered to 28 (Roman-González et al., 2017). Cronbach's Alpfa reliability as the internal consistency of the CTt is a=0.793, which is regarded to be high reliability (Nunnally & Bernstein, 1994). With a range of p=0.16 (item 23, very high difficulty) to p=0.96 (item 1, very low difficulty), the average rate of achievement for the 28 items is p=0.59 (medium difficulty).

The test consists of 28 multiple-choice questions. To introduce students with the test questions and the characters used in them, they are provided with three example questions along with their right responses prior to the exam. Seven basic CT concepts are covered in the questions on the CTt. Simple functions (4 questions), basic sequence structure (4 questions), while type loop (4 questions), repeat... times type loop (4 questions), simple (if) decision statement (4 questions), repeat until type loop (4 questions), and complex (if-else) decision statement (4 questions).

The researcher adopted the CTt from English to Turkish that can be used by students in secondary school across all grade levels by organizing the test according to the recommendations of two faculty members with expertise in science education, one faculty member with expertise in educational sciences, and a language expert. Furthermore, to ensure the test's language validity, it was translated from English to Turkish and back again from Turkish to English. After the appropriate verifications were completed, the test was then given its final version (see APPENDIX A).

The adapted test was applied to 353 students studying in the fifth, sixth, seventh, and eighth grades in two secondary schools in Istanbul for validity and reliability studies. The data was collected in the fall semester of the 2022-2023 academic year. The SPSS program was used to analyze the data collected from the student group. As a result of the adaptation of the test to Turkish, while the original version of the test had 28 questions, the number of questions was reduced to 25 in the Turkish version. Cronbach Alpha reliability coefficient of the entire adapted test is 0.832.

Normality distribution of CTt is given in Figure 3.1.

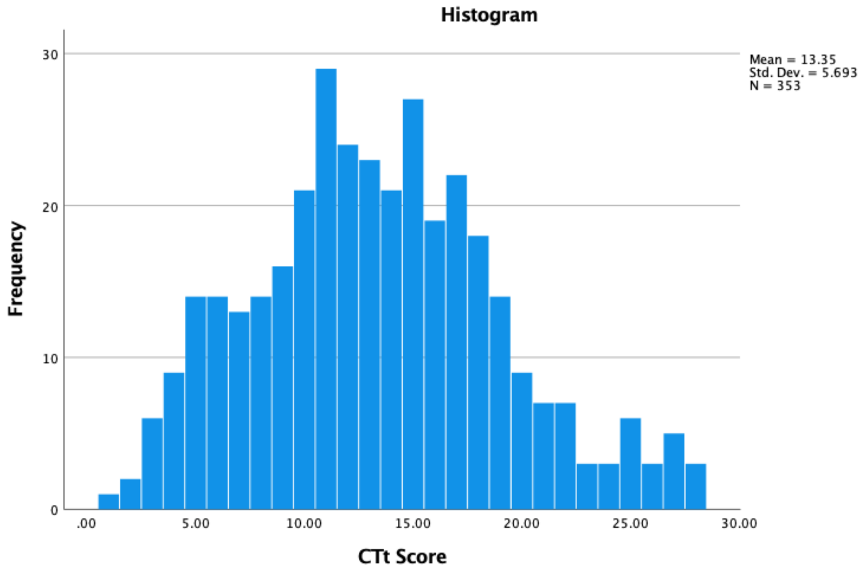


Figure 3.1 Histogram of the CTt Score

A histogram illustrating the distribution of the CTt score throughout the sample is shown in figure 3.1. The distribution stated above, as can be observed, fits the normal curve quite well because according to Hair et al. (2010), Skewness values between -1 to +1 indicate a normal distribution if the sample size (n>300). Corrected item-total correlations of CTt are given in Table 3.4.

Table 3.4 Corrected Item Total Correlations

# of Questions	Corrected Item Total Correlation	# of Questions	Corrected Item Total Correlation
1	.325	15	.129
2	.371	16	.415
3	.462	17	.326
4	.470	18	.370
5	.563	19	.349
6	.406	20	.385

Table 3.4 Corrected Item Total Correlations (continued…)

7	.393	21	.303
8	.325	22	.343
9	.423	23	.406
10	.377	24	.358
11	.383	25	.303
12	.079	26	.281
13	.349	27	.252
14	.337	28	.328

#: Number

Since corrected item-total correlations with scores lower than .30 (items numbered 12, 15, 26, and 27) reduce the reliability of the test, it was predicted that removing them from the test would increase the reliability of the test. However, these items were not removed from the test without performing factor analysis.

Table 3.5 Item Statistics

# of Item	**Item Difficulty**	**Item Discrimination**	**# of Item**	**Item Difficulty**	**Item Discrimination**
1	.82	0.35	15	.36	0.13
2	.70	0.53	16	.31	0.55
3	.60	0.69	17	.40	0.53
4	.38	0.64	18	.42	0.61
5	.65	0.81	19	.35	0.47

Table 3.5 Item Statistics (continued…)

6	.67	0.56	20	.43	0.58
7	.53	0.57	21	.49	0.49
8	.44	0.49	22	.25	0.43
9	.55	0.62	23	.30	0.52
10	.55	0.59	24	.46	0.55
11	.64	0.48	25	.32	0.43
12	.26	-0.01	26	.47	0.40
13	.52	0.48	27	.48	0.41
14	.48	0.46	28	.51	0.45

The second challenging question of the test is the twelfth question, and its item discrimination index is negative. Therefore, the children in the higher score group and the lower score group are not significantly distinguished from one another. The difficulty index of the fifteenth item was determined to be .36. Item number fifteen is categorized as a medium-difficulty item. However, it was discovered that the item's discrimination index was .13. Despite being about a medium hard item, this one does not successfully distinguish between students in the upper and lower score groups. It was thought that taking them out of the test would improve test reliability.

Descriptive statistics for the items before questions 1,12, and 15 are removed from the test and are given in Table 3.6.

Table 3.6 Descriptive Statistics

# of students	Mean	SD	Variance	Lowest	Highest	Kurtosis	Skewness
353	13.35	5.69	32.40	1	28	-.22	.30

Descriptive statistics for the items after questions 1,12, and 15 are removed from the test are given in Table 3.7.

Table 3.7 Descriptive Statistics

# of students	Mean	SD	Variance	Lowest	Highest	Kurtosis	Skewness
353	11.90	5.42	29.38	1	25	-.311	.303

In line with the validity studies of CTt, the content validity and construct validity of the test were examined. To test the construct validity of the test, the Kaiser-Meyer-Olkin (KMO) value and Bartlett Sphericity Test were applied to the data obtained from the group. KMO value was calculated as .798. In addition, Bartlett's Test of Sphericity was found to be significant [X=2134.756 p<.001]. When considering whether the data is appropriate for factor analysis, the KMO value is greater than 0.60 and the Bartlett Test demonstrates significance (Büyüköztürk et al., 2020). While the mean of the original test was 16.38, the mean of the Turkish adaptation was 11.9. The reason for this difference may be cultural factors, the education they receive at school or the activities they do in their free time.

When considering the test was modified for use in secondary school, exploratory factor analysis was used to analyze the data and determine the type of factor structure that emerged from the information gathered from the student group that can be seen in figure 3.2.

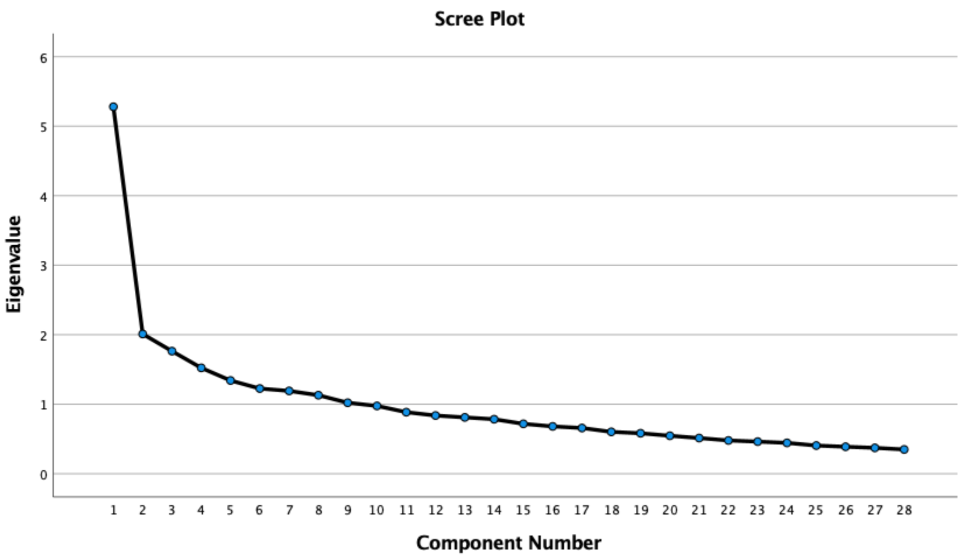


Figure 3.2 Exploratory factor analysis eigenvalue Scree plot for CTt

The Varimax orthogonal rotation technique was used to apply principal component analysis to the data (Büyüköztürk et al., 2020). Following examination of the first analysis results, it was observed that the test's eigenvalue was gathered in 9 elements that had values greater than one. Nine significant factors with significant decreases can be seen in Table 8. that illustrates the eigenvalues of the factors identified by exploratory factor analysis. The final analysis restricted the number of components to seven that were considered conceptually appropriate in considering the item contents. The examination revealed that there were now just 25 items on the test after three items (number 1, number 12, and number 15) with a factor loading below .40 had been removed from the test. To ensure the construct validity of the test, the 26th and 27th items, whose corrected item correlation was less than .30, remained in the test because their factor loadings were equally distributed and greater than .40.

After conceptual analysis, items were collected into seven separate factors. These factors are “Loops-Repeat Times”, “If/Else-Complex Conditional”, “Basic Directions and Sequences”, “If-Simple Conditional”, “Loops Repeat Until”, “Simple Functions”, and “While Conditional”. The "Basic Directions and Sequences" contains three components (numbered 2,3,4), 4 items in the “Loops-Repeat Times” (numbered 5,6,7,8), 3 items in the “Loops Repeat Until” (numbered 9,10,11), 4 items in the “If-Simple Conditional” (numbered 13,14,15,16), 4 items in the “If/Else-Complex Conditional” (numbered 17,18,19,20), 4 items in the

"While Conditional" (numbered 21,22,23,24) and 4 items in the "Simple Functions" (numbered 25,26,27,28). The eigenvalues for these factors were determined as follows: Basic Directions and Sequences factor accounts for 20.42% of the total variance, Simple Functions factor accounts for 28.30%; If/Else-Complex Conditional factor accounts for 35.26%; While Conditional factor accounts for 41.17%; Loops-Repeat Times factor accounts for 46.08%; If-Simple Conditional factor accounts for 50.73%; and Loops Repeat Until factor accounts for 55.00%. These seven factors together explain 55% of the total variance.

Table 3.8 Principal Component Analysis

# of questions	Component						
	Basic Directions and Sequences	**Simple Functions**	**If/Else-Complex Conditional**	**While Conditional**	**Loops - Repeat Times**	**If-Simple Conditional**	**Loops Repeat Until**
2	.663						
3	.722						
4	.654						
5					.577		
6					.686		
7					.614		
8					.744		
9							.827
10							.560
11							.619
13						.744	

Table 3.8 Principal Component Analysis (continued…)

14						.647	
16						.674	
17			.560				
18			.667				
19			.511				
20			.667				
21				.551			
22				.756			
23				.782			
24				.469			
25		.588					
26		.745					
27		.759					
28		.720					

As seen in Table 3.8. item factor loadings in the basic directions and sequences factor varied between .654-.722, in the simple functions factor .588-.759, in the if/else-complex conditional factor .511-.667, in the while conditional factor .469-.782, in the loops-repeat times factor .577-.744, in the if-simple conditional factor .647-.744, and the loops repeat until factor varied between .560-.827. It was observed that the item factor loadings of the entire test varied between .484-.835.

Question numbered 5 addresses basic directions and sequences for .390, but depending on the original of the test, question numbered 5 addresses basic directions and sequences. Question 21 addresses basic directions and sequences for .413, but according to the original of the test, question 21 addresses basic directions and sequences, therefore these questions were not removed from the test.

After construct validity, to test the internal consistency, the Cronbach's alpha coefficient of the CTt was investigated within the scope of reliability studies. The reliability coefficient of the entire test is .832. The reliability coefficients of the sub-factors were found to be: .598, .696, .586, .580, .592, .613, and .687 for basic directions and sequences, loops-repeat times, loops repeat until, if-simple conditional, if/else-complex conditional, while conditional, simple functions, and respectively. This shows that the test is a valid and reliable measurement tool.

3.3.2 Force and Motion Test (FMKt)

The force and motion knowledge test (FMKt) was developed by Çimentepe (2019) to determine sixth-grade students' knowledge levels about the force and motion unit. 40 multiple choice questions were created at the beginning of the research for all concepts in the Force and Motion unit. For the validity and reliability of the knowledge test, an application was made with 83 students. As a result of the analysis made among the 40 questions prepared for the Force and Motion unit, the 25 questions that worked best were determined. The reliability of the final version of the knowledge test (see APPENDIX B), consisting of 25 questions, was calculated as .87, the item discrimination index was .49 and the item difficulty index was .49.

Table 3.9 Distribution of Questions Associated Science Objectives

Number of Objectives	Associated Science Objectives	Number of the Questions
F.6.3.1.1	Showing the direction and magnitude of the force acting on an object by drawing it.	6, 8, 10, 12, 15, 18, 19
F.6.3.1.2	Observing by trying more than one force acting on an object	3, 5, 9, 16, 20, 23
	The emphasis is on the resultant of forces in the same direction. The resultant of forces with different directions cannot be entered.	

Table 3.9 Distribution of Questions Associated Science Objectives (continued…)

F.6.3.1.3	Comparing balanced and unbalanced forces by observing the motion of objects	2, 4
F.6.3.2.1	Defines speed and expresses its unit.	1, 7, 11, 22, 25
	a. Meter/second (m/sec.) and kilometer/hour (km/h) are considered as speed units.	
	b. The concepts of displacement and velocity are not included.	
	c. Mathematical relations are not included	
	d. There is no unit conversion.	
F.6.3.2.2	Showing the relationship between distance, time, and speed on a graph	13, 14, 17, 21, 24

3.3.3 Activity Evaluation Form

The activity evaluation form was prepared by the researcher to determine students' opinions about the course activities integrated with Scratch. Activities were carried out for 4 weeks and an activity evaluation form consisting of open-ended questions was applied at the end of the research. The activity evaluation form was prepared as open-ended to examine students' opinions in depth. According to Yıldırım & Şimşek (2005), when collecting more in-depth information on the topic, open-ended questions allow the researcher a great deal of flexibility. After the items of the form were prepared, they were checked by two experts specialized in science education and two science teachers. There are 5 questions in the event opinion form before it is submitted to expert opinion. However, the number of questions was increased to 7 because experts thought that the small number of questions might cause difficulties in providing information about the activities carried out on Scratch. The form was given its final shape in line with the recommendations of the experts. The final version of the activity evaluation form contains 7 questions (see APPENDIX C). These questions are: "What is Scratch?", "What do you do with

Scratch?", "In what other areas of your life can you use Scratch?", "How did you find the idea of the project in Scratch? How did you start doing the project?", " Did you have difficulty creating a project in Scratch? What did you do when you had difficulty in creating the project?", "What do you like and dislike about Scratch? What features do you want to change or stay the same?" and "What other things do you like to do related to technology?". Students were asked to write their answers to these questions. The answers given by the experimental group students to the open-ended activity evaluation form were analyzed by content analysis and then interpreted. Twenty percent of the qualitative data obtained from the activity evaluation form was re-analyzed by another researcher. The reliability coefficient between the scores given by the two evaluators was found to be .94. The researcher also reanalyzed these data 6 months later and found the reliability coefficient for this scoring to be .90. Thus, scoring reliability was ensured.

3.4 Integrating Scratch into the course as a Block-Based Programming Environment

There are five objectives in the force and motion unit in the sixth-grade science curriculum. In this study, lesson plans integrated with Scratch were developed for each objective in the program. Lesson plans prepared for the planned execution of the lessons are presented as an attachment (see APPENDIX E). Lessons were followed in a planned manner based on these plans. Additionally, the worksheets prepared for this course are presented in the appendixes of the lesson plans (see APPENDIX E).

After the arrangement of the lesson plans was completed, the lesson plans were presented to two faculty members who are experts in science education and two science teachers to obtain expert opinions. The lesson plans were finalized in line with the opinions of the experts. The prepared lesson plans integrated with Scratch are suitable for the sixth-grade level and the Science Education curriculum.

Examples of the pictures taken during the application in the experimental group are presented in Appendix F. The attached images are images taken to keep the identity of the students private.The distribution of Scratch programming activities according to the objectives in the science curriculum is given in Table 3.10.

Table 3.10 Distribution of Scratch Programming Activities According to Objectives

Activity Name	Objectives Related to the Activity
1. Let's move a motionless object	1. Showing the direction and magnitude of the force acting on an object by drawing it.
2. Who wins the tug-of-war game?	2. Observing by trying more than one force acting on an object
3. Balanced or Unbalanced Force?	3. Comparing balanced and unbalanced forces by observing the motion of objects
4. I'm Making a Speed Calculator	4. Defines speed and expresses its unit
5. I Design Graphics	5. Showing the relationship between distance, time and speed on a graph

The sample activities prepared in Scratch were designed by paying attention to the CT concepts addressed in the CTt used in the study. In addition, Scratch projects on that subject were developed by the researcher for each course, paying attention to science objectives.

The distribution of Scratch programming activities according to the addressed CT concepts is given in Table 3.11.

Table 3.11 Distribution of Scratch Programming Activities According to Computational Concepts

Activity Name	Computational Thinking Concepts Addressed
1. Let's move a motionless object	Basic Directions and Sequences
2. Who wins the tug-of-war game?	Basic Directions and Sequences If-Simple Conditional Simple Functions

Table 3.11 Distribution of Scratch Programming Activities According to Computational Concepts (continued…)

	Loops-Repeat Times Loops Repeat Until
3. Balanced or Unbalanced Force?	Basic Directions and Sequences If-Simple Conditional Simple Functions
4. I'm Making a Speed Calculator	Basic Directions and Sequences Loops Repeat Times Simple Functions While Conditional If-Else Complex Conditional
5. I Design Graphics	Basic Directions and Sequences Loops Repeat Times

Scratch activities prepared within the scope of lesson plans and used as a sample activity in the study are given below. The researcher gave these activities as examples. Under the condition that the subject of the project was related to the topics presented in the course, students were allowed to develop their projects during the course. Students used the scene in Scratch, puppets, added sounds to their projects and used code blocks as they wished.

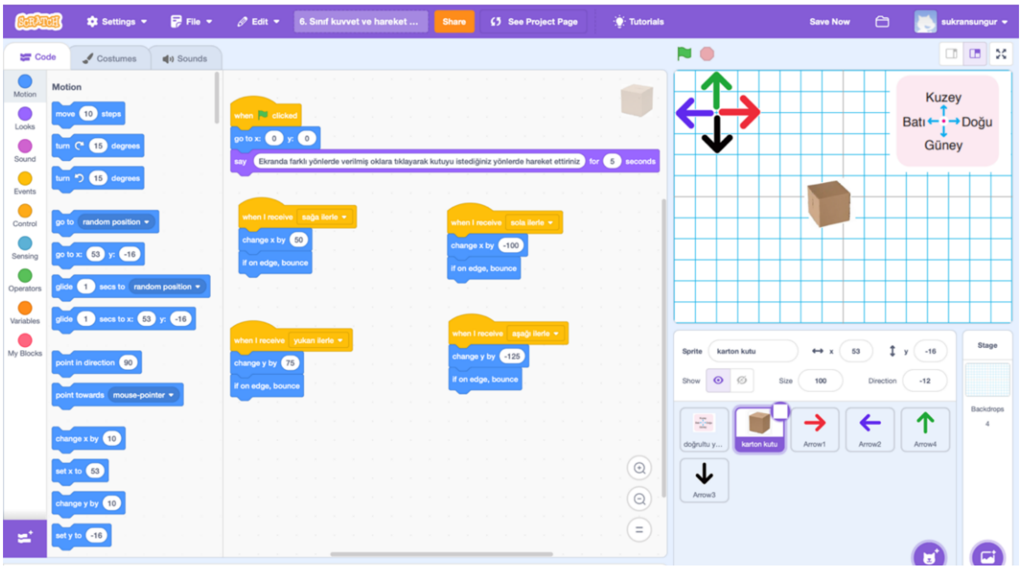


Figure 3.3 Scratch Activity: Let's move a motionless object

In this activity, students understood the direction of the force acting on an object by coding. The object moved in the north, south, east, and west directions. According to this activity, when you click on the colored arrows in the image, the object moves in that direction. For example, when the red arrow is clicked, the object moves to the east, and when the green arrow is clicked, the object moves to the north. Since this activity is the first activity, it addresses the basic directions and sequence sub-dimension of CT concepts. Additionally, with this activity, students learned the direction and magnitude of the force acting on an object, which is the objective of the course.

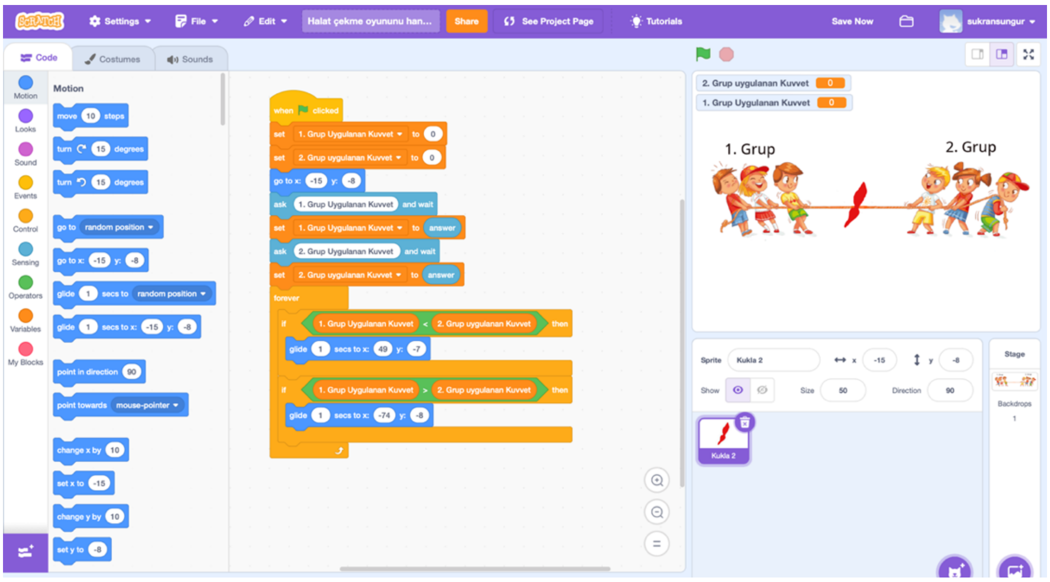


Figure 3.4 Scratch Activity: Who wins the tug-of-war game?

With this activity in Scratch students experienced more than one force act on an object by coding. There are two groups in this activity, and whichever group applies more force to the rope wins the tug-of-war game. Two questions are asked depending on the activity prepared here. The first question asks about the force applied by the first group, and the second question asks about the force applied by the second group. In this activity, students determine the force values for these two groups. As a result, they observe which group won the tug-of-war game according to the force values they determined. With the code blocks used in this activity, basic directions and sequences, if-simple conditional, simple functions, loops-repeat times, loops repeat until sub-dimensions of CT were addressed. At the same time, students observed how more than one force acting on an object affects the object.

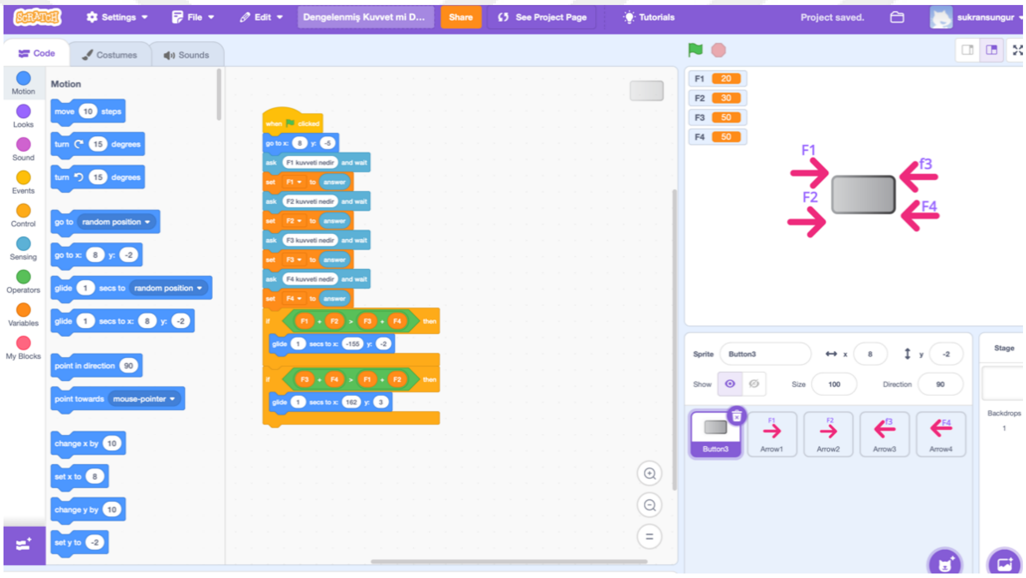


Figure 3.5 Scratch Activity: Balanced or Unbalanced Force?

Through this activity, students discovered when more than one force acts on an object, the resulting force can either cause balanced or unbalanced forces. Here, more than one force value is acting on an object. These force values are asked to the students, and because of the force values determined by the students, it appears on the screen whether the object is under the influence of balanced or unbalanced forces. In this activity, basic directions and sequences, if-simple conditional, and simple functions sub-dimensions of CT were mentioned. Additionally, students learned balanced and unbalanced forces with code blocks.

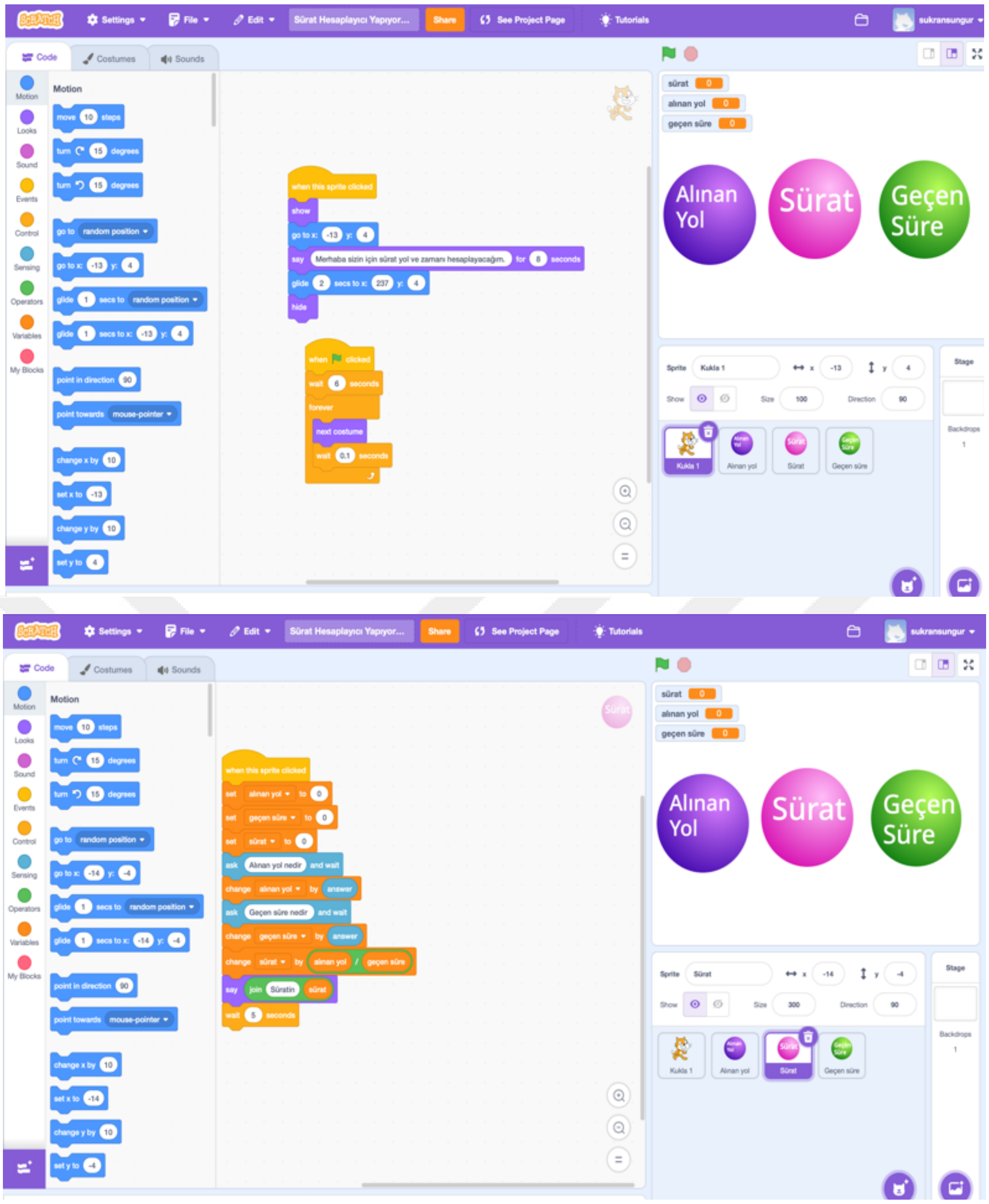


Figure 3.6 Scratch Activity: I'm Making a Speed Calculator

In this activity, students define speed and express its unit by coding. In this activity, there are speed, distance, and time costumes. Students are asked about the distance and the time. The speed calculator calculates the speed according to the distance traveled and time values determined by the students. With the code blocks used in this activity, basic directions and sequences, if-else complex conditional, simple functions, loops-repeat times, while conditional sub-dimensions of CT were addressed. At the same time, students learn how to calculate speed with code blocks.

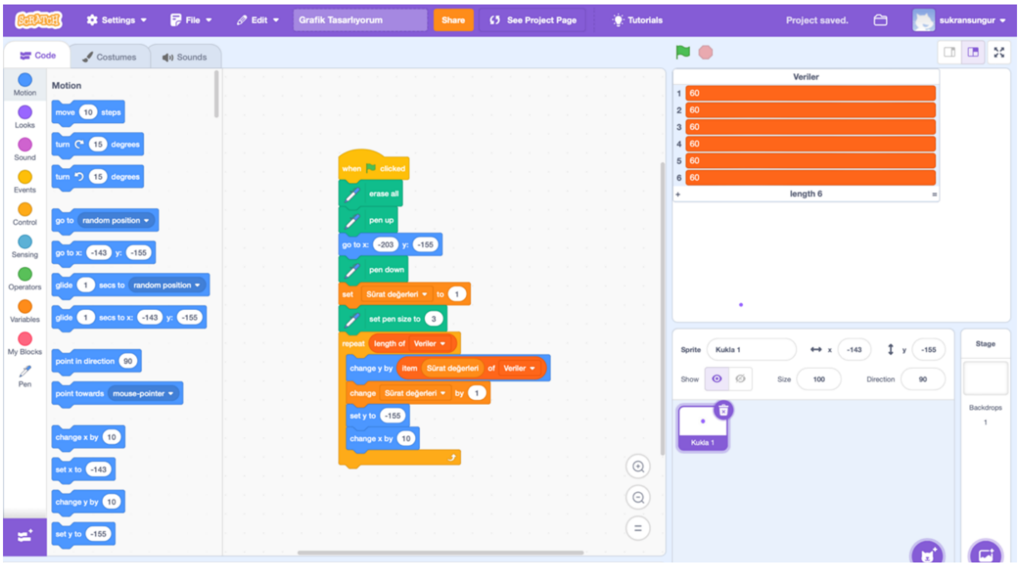


Figure 3.7 Scratch Activity: I Design Graphics

With this activity in Scratch students, they learn to draw the graph of a constant-speed object with code blocks. Six-speed values were given in this activity. These speed values are equal, and the speed graph of the object is drawn with the code blocks prepared according to these speed values. With the code blocks used in this activity, basic directions and sequences, and loops-repeat times sub-dimensions of CT were addressed.

3.5 Procedure

The main implementation of this study was carried out in the "Force and Motion" unit of the sixth-grade science course in the spring semester of the 2022-2023 academic year. The research lasted a total of six weeks. The achievement of force and motion and computational thinking of both groups were determined before and after the application. The pre-tests were used in the research, and then the experimental study was applied. Furthermore, comparisons were made between groups by looking at how the groups changed within themselves.

According to this situation;

1. Before starting the implementation of the research, the necessary permissions were obtained from the Yıldız Technical University Social and Humanities Research Ethics Board. In the decision dated 03.10.2022 and meeting number 2022.10, the ethics committee allowed the research.

2. After obtaining permission from the University's Ethics Committee, the necessary permissions were obtained from the MoNE (see APPENDIX D).
3. Table 12 provides an extensive overview of the curriculum that was followed by the control and experimental groups during the study. The program's suitability for the sixth-grade Ministry of Education curriculum was carefully considered during the program's development, and the groups' science teacher opinion was obtained.
4. Two sixth-grade classes were randomly assigned to the control and experimental groups. These two classes were selected with convenience sampling as two classes where the same science teacher taught.
5. The research was conducted in the Science course in order not to disrupt education.
6. Two groups were used in the research: a control and an experimental group. During the research, the Force and Motion unit was taught using Scratch in the experimental group, while the MoNE's recommended approach (supported by examples and activities from the Science 6 textbook) was followed in the control group.
7. In the experimental group, lessons were taught using 5 different lesson plans (see APPENDIX E) and activities prepared by the researcher regarding the subjects planned to be taught in accordance with the subject content of the 6th grade Science Course Curriculum (MoNE, 2018). In the control group, lessons were taught using the texts, experimental activities, and questions about the activities in the Science 6 Coursebook (Aydin et al., 2022), as prescribed by the MoNE.
8. In both groups, the lessons were conducted by the same teacher who was a graduate of the Science Teaching Program.
9. In the experimental group, science lessons were conducted in the computer laboratory class, while in the control group, lessons were conducted in the students' classrooms.
10. Each lesson lasted 40 minutes. The pre-tests and post-tests of both groups are the same. Differently, the activity evaluation form was applied to the experimental group students after the application.

Table 3.12 Curriculum that was followed by the experimental and control groups during the study

Weeks	Actions
1st Week	Pre-tests were administered to the groups
2nd Week	The topic of "forces acting on an object" was discussed
3rd Week	The topic of "balanced and unbalanced forces" was discussed
4th Week	The topic of "constant speed motion" was discussed
5th Week	The topic of "graphs on constant speed motion" was discussed
6th Week	Post-tests were administered to the groups

3.5.1 Procedures Followed in the Experimental Group

The integration of Scratch programming activities into the subjects of the force and motion unit started with studies prepared as examples in Scratch for students. When the sample studies were shared with the students, they were first asked to complete the tasks individually by following the instructions in the study given to them. A volunteer student was asked to show the completed instruction on the board to the whole class. Then, the prepared worksheets were given to the students after the class was divided into groups of four. A screenshot of the distributed worksheet was opened on the screen and shown to the students. Here, students were asked to answer questions as a group. Meanwhile, the teacher walked around the classroom and helped students if they had questions. It was said that after the study was completed, one student from each group would be given the right to speak. Creating a discussion-based environment in the group is thought to improve students' interest in the course. Moreover, students may learn from each other's ideas and correct each other's mistakes by discussing things they think are incorrect. After students work on the project designed by the researcher on the Scratch program, the students are asked to design a project related to the subject of the course in Scratch. Students are asked to share the work they have prepared with their teacher and friends. After students share their work with the teacher, the code blocks of the work prepared by the students are checked and the students are asked why they used this code block. Depending on the answer given by the student, another student is asked: "Your friend thinks this way, what do you think about it?" In this way, a discussion environment is provided in the classroom.

The teacher guided during the course, helping the students find the correct answer by giving them clues, rather than directly correcting their mistakes during the process. While students were designing projects in Scratch, they were given as much freedom as possible during the process. At the end of each topic, the questions in the worksheets prepared by the researcher were solved with the students.

In addition, the teacher solved the questions suggested by the MoNE on the smart board with the students at the end of the unit.

3.5.2 Procedures Followed in the Control Group

No intervention was performed in the control group. The teacher of the course conducted the lesson by following the flow of the current program.

While planning the course of the lessons in the control group, the school teacher was interviewed and information was obtained about the course process. The teacher stated that in this lesson, they did the activities in the book distributed by the MoNE (Aydin et al., 2022) and solved sample questions on the subjects covered in the Science course. Lessons were taught without any intervention as prescribed by the MoNE.

In this group, the teaching of the subject was carried out by following the Science 6 Course Book and using the exercises and examples given in the book. The activities in the book were carried out in the order of the book. The equipment for the activities in the book is easily available.

Throughout the teaching, students took notes in their books with the answers to the questions in the book. The discussions at the end of the activities were held as a collective class discussion, and each student was ensured active participation in the process. At the end of each topic, sample questions prepared by the researcher on the topic, which were the same as the experimental group, were answered by the students.

In addition, the teacher solved the questions suggested by the MoNE on the smart board with the students at the end of the unit.

3.6 Data Analysis

The SPSS statistical program was used to evaluate quantitative data obtained from the tests used in the research process. Before data analysis, descriptive statistics like homogeneity, kurtosis, skewness coefficients, and normality were assessed for CTt and FMKt each. The analysis of the FMKt, CTt and the activity evaluation form are described below.

3.6.1 Computational Thinking Test (CTt)

The pre and post-test of the CTt, adapted to Turkish by the researcher, taken from the control and experimental groups were analyzed. CTt were analyzed to answer the first and second research questions. To use the t-test in groups, it was first checked whether the CTt scores within the group were normally distributed or not. Levene's test was used to determine if the scores for each group's dependent variable were homogeneous in terms of variance. The variances in this test are considered homogeneous if the p-value is more than .05 ($p>.05$) (van Aert et al., 2016). The Shapiro-Wilk test was performed to statistically verify normality because this test is the more appropriate method for small sample sizes ($n<50$ samples).

3.6.2 Force and Motion Knowledge Test (FMKt)

The pre and post-test of the force and motion knowledge test taken from the control and experimental groups were analyzed. FMKt was analyzed to answer the third and fourth research questions. To use the t-test in groups, it was first checked whether the FMKt scores within the groups were normally distributed or not. Levene's test was used to determine if the scores for each group's dependent variable were homogeneous in terms of variance. To statistically confirm normality, the Shapiro-Wilk test was run.

3.6.3 Activity Evaluation Form

In the analysis of qualitative data, content analysis was performed. Content analysis helps identify, organize and interpret data (Creswell-John & Creswell-David, 2017). The steps of content analysis consist of collecting data, coding the obtained data, creating categories and themes from the codes, and tabulating the data. Every analysis unit has a code assigned to it through analysis. Based on the similarities and differences between the codes, categories are created. Based on the meanings each category has, themes are developed from them. Lastly, a visual representation of the data is made (James-McMillan & Schumacher, 2010). In this research, content analysis was done on the data obtained from the open-ended activity evaluation form. Frequency percentages of student responses were found. Examples of student responses are presented along with the numbers assigned to each student. The results obtained are given in tables.

4
RESULTS

The findings of the data analysis conducted to investigate the research questions are presented in this chapter. Firstly, CTt results, secondly, FMKt results, and lastly results for qualitative data based on the activity evaluation form are given.

4.1 Results for Computational Thinking Test (CTt)

The data obtained as a result of the pre and post-test application of the CTt for experimental and control groups are given in Table 4.1.

Table 4.1 Descriptive statistics of the Computational Thinking Test

Data Collection Time	Group	N	M	SD	Min	Max	Range
Pre-test	Experimental Group	23	11.0	2.12	7	15	8
	Control Group	28	11.8	2.47	7	17	10
Post-test	Experimental Group	23	17.0	2.98	12	23	11
	Control Group	28	12.0	2.91	6	17	11

The pre and post-test results for students in the experimental and control groups on CTt are presented in Table 4.1. Students in the experimental group scored 11.0 on the CTt, compared to 11.8 in the control group's pre-test mean score. With a 0.8 difference in mean score, the control group outperforms the experimental group. The experimental group's mean score on the CTt was 17.0 for the post-test, whereas the control group's score was 12.0. There is a 5.0 point difference in the post-test mean score between the experimental and control groups on CTt.

The pre-test and post-test of the CTt taken from the experimental and control groups were analyzed. To use the t-test in groups, it was first checked whether the CTt scores within the group were normally distributed or not. Levene's test was used to determine if the scores for each group's dependent variable were homogeneous in terms of variance. The variances in this test are considered homogeneous if the p-value is more than .05 ($p>.05$) (van Aert et al., 2016). The Shapiro-Wilk test was performed to statistically verify normality because this test is the more suitable method for small sample sizes ($n<50$ samples). The p-value was found to be significant ($p>.05$).

Table 4.2 The normality table of the experimental group

Data Collection Time	N	Min	Max	M	SD	Skewness	Kurtosis
Pre-test	23	7	15	11.04	2.12	-.062	-.682
Post-test	23	12	23	17.00	2.98	.270	-.654

Table 4.2 shows the normality table of the experimental group's pre and post-test results for the CTt. The paired samples t-test was used to calculate the difference between the pre and post-test scores. The CTt scores within the group were checked to see if they showed a normal distribution before applying the paired samples t-test in the dependent group. Since the pre and post-test of the experimental group show normal distribution, the paired sample t-test was used to answer the first research question which is "Are there any significant differences in the computational thinking of sixth-grade students after the intervention?".

Table 4.3 The normality table of the control group

Data Collection Time	N	Min	Max	M	SD	Skewness	Kurtosis
Pre-test	28	7	17	11.85	2.47	.018	-.370
Post-test	28	6	17	12.03	2.91	-.329	-.714

The normality table of the pre and post-test results for the CTt of the control group is given in Table 4.3. Given that the experimental and control groups' pre and post-tests showed a normal distribution, an independent sample t-test was used to answer the second research question which is “Are there any significant differences in the computational thinking of sixth-grade students between the control and experimental group?”. It was required to determine whether there was a substantial difference between the experimental and control groups' pre-test results to prevent the effect of the current difference between the groups' pre-test scores. An independent sample test was used because there was not a significant difference between the experimental and control groups' pre-test results (t=-1.245, p>.05).

The experimental groups were given CTt with multiple choice both before and after the treatments. Throughout the research, activities were conducted to find out how much using block-based programming environments could change students' prior knowledge of CT. In other words, the question of whether the teaching strategy was enough to change CT was answered by comparing the pre and post-test results. The students in the experimental group showed an increase in scores when comparing their pre and post-test results. Since the pre and post-test of an experimental group show normal distribution, the paired sample t-test was used to calculate the difference between the pre and post-test scores. The paired samples t-test of the experimental group is presented in Table 4.4.

Table 4.4 Results of the paired sample t-test of the experimental group

Experimental Group Tests	N	M	SD	t	P
Pre-test	23	11.0	2.12	-25.807	<.001
Post-test	23	17.0	2.98		

The pre and post-test means of the experimental group were found to be 11.0 and 17.0 respectively. As a result of the paired sample t-test, a statistical difference was found between these two results, t=-25.807, p<.05. This supports that the pre and post-test means of the experimental group are statistically different from each other. The score of the pre-test CTt is lower (M=11.0) than the score of the post-test

(M=17.0). Therefore, it can be said that science lessons integrated with block-based programming environments contribute positively to students' CT.

The force and motion unit was taught using Scratch activities in the experimental group, while the control group received instruction using the MoNE's recommended teaching method. The experimental and control groups were given CTt with multiple choice as pre-test and post-test. Throughout the research, activities were carried out to investigate whether the use of block-based programming environments would make a difference in students' CT between the experimental and control groups. The students in the experimental group showed an increase in scores when comparing their pre and post-test results. Since the pre and post-test of the experimental and control groups show normal distribution, the independent sample t-test was used to calculate the difference between the experimental and control groups. Before using the independent sample t-test, it was first determined whether the pre-test scores for the experimental and control groups were equal. The pre-test mean scores of the experimental and control groups for CTt were found to be 11.0 and 11.8 respectively. As a result of the independent sample t-test, no statistical difference was found between the pre-tests of experimental and control groups, t=-1.245, p>.05. This statistically supports that the pre-test mean scores of the experimental and control groups are not different from each other. The independent samples t-test of the experimental and control groups is presented in Table 4.5.

Table 4.5 Results of the independent sample t-test of the experimental and control groups' post-test

Group	N	M	SD	t	p
Experimental Group	23	17	2.98	20.950	<.001
Control Group	28	12	2.91		

The post-test mean score of the experimental and control groups for CTt were found to be 17.0 and 12.0 respectively. The independent sample t-test showed a statistically significant difference (t=20.950, p<.05) between the experimental and control groups post-test . According to this, the pre and post-test difference averages of the experimental and control groups are shown to be statistically different from each other. The score of the experimental group is higher (M=17.0) than the score of the control group (M=12.0). Thus, it can be stated that science classes that integrate a block-based programming environment have a positive impact on students' CT.

4.2 Results for Force and Motion Knowledge Test (FMKt)

The data obtained as a result of the pre and post-test application of the FMKt for experimental and control groups are given in Table 4.6.

Table 4.6 Descriptive statistics of the Force and Motion Test

Data Collection Time	Group	N	M	SD	Min	Max	Range
Pre-test	Experimental Group	23	5.50	2.20	2	10	8
	Control Group	28	5.75	1.93	3	10	7
Post-test	Experimental Group	23	15.20	3.70	9	22	13
	Control Group	28	13.39	2.83	8	20	12

The pre and post-test results for students in the experimental and control groups on FMKt are presented in Table 4.6. Students in the control group scored 5.75 on the FMKt, compared to 5.5 in the experimental group's pre-test mean score. With a 0.25 difference in mean score, the control group outperforms the experimental group. The experimental group's mean score on the FMKt was 15.2 for the post-test, whereas the control group's score was 13.39. There is a 1.81 point difference in the post-test mean score between the experimental and control groups on FMKt.

The pre and post-test of the FMKt taken from the experimental and control groups were analyzed. To use the t-test in groups, it was first checked whether the force and motion knowledge test scores within the groups were normally distributed or not. Levene's test was used to determine if the scores for each group's dependent variable were homogeneous in terms of variance. To statistically confirm normality, the Shapiro-Wilk test was run. The p-value was found to be significant (p>.05).

Table 4.7 The normality table of the experimental group

Data Collection Time	N	Min	Max	M	SD	Skewness	Kurtosis
Pre-test	23	2	10	5.478	2.273	.387	-.849
Post-test	23	9	22	15.260	3.719	.251	-.872

The normality table of the pre and post-test results for the FMKt of the experimental group is given in Table 4.7. When considering that the normal distribution assumption was satisfied, the experimental group's pre and post-test on FMKt were analyzed using a paired sample t-test to answer the third research question which is "Are there any significant differences in the achievement of force and motion unit of sixth-grade students after the intervention?".

Table 4.8 The normality table of the control group

	N	Min	Max	M	SD	Skewness	Kurtosis
Pre-test	28	3	10	5.75	1.93	.645	-.001
Post-test	28	8	20	13.39	2.83	.116	-.356

The normality table of the pre and post-test results for FMKt of the control group is given in Table 4.8. Since the groups are normally distributed, an independent sample t-test can be applied to answer the fourth research question. To prevent the impact of the current difference between the pre-test scores of the groups, it was necessary to determine whether there was a significant difference between the experimental and control groups' pre-test results. Since there was no significant difference between the pre-tests of the experimental and control groups, t=-.461, p>.05 an independent sample test was applied to answer the fourth research question which is “Are there any significant differences in the achievement of force and motion unit of sixth-grade students between the control and experimental group?”.

The experimental groups were given FMKt with multiple choice both before and after the treatments. Throughout the research, activities were conducted to find out how much using block-based programming environments could change students' prior knowledge of force and motion. In other words, the question of whether the teaching strategy was enough to change force and motion knowledge was answered by comparing the pre and post-test results. The students in the experimental group showed an increase in scores when comparing their pre and post-test results. Since the pre and post-test of an experimental group show normal distribution, the paired sample t-test was used to calculate the difference between the pre and post-test scores. The paired samples t-test is presented in Table 4.9.

Table 4.9 Results of the paired sample t-test of the experimental group

Experimental Group Tests	**N**	**M**	**SD**	**t**	**p**
Pre-test	23	5.50	2.20	-9.488	<.001
Post-test	23	15.20	3.70		

The pre and post-test means of the experimental group were found to be 5.5 and 15.2 respectively. As a result of the paired sample t-test, a statistical difference was found between these two results, $t=-9.488$, $p<.05$. This supports that the pre and post-test means of the experimental group are statistically different from each other. The score of the pre-test FMKt is lower ($M=5.5$) than the score of the post-test ($M=15.2$). Therefore, it can be said that science lessons integrated with block-based programming environments contribute positively to students' knowledge of force and motion.

The Force and Motion unit was taught using Scratch activities in the experimental group, while the control group received instruction using the Ministry of National Education's recommended teaching method. The experimental and control groups were given FMKt with multiple choice as pre-test and post-test. Throughout the research, activities were carried out to investigate whether the use of Scratch integrated activities would make a difference in students' knowledge of force and motion between the experimental and control groups. The students in the experimental and control groups showed an increase in scores when comparing their pre and post-test results. Since the pre and post-test of the experimental and control groups show normal distribution, the independent sample t-test was used to calculate the difference between the experimental and control groups. Before using the independent sample t-test, it was first determined whether the pre-test scores for the experimental and control groups were equal. The pretest mean scores of the experimental and control groups for FMKt were found to be 5.5 and 5.75 respectively. As a result of the independent sample t-test, no statistical difference was found between the pre-tests of the experimental and control groups, $t=-.461$, $p>.05$. This supports that the pre-test mean scores of the experimental and control groups are not statistically different from each other. The independent samples t-test of the experimental and control groups is presented in Table 4.10.

Table 4.10 Results of the independent sample t-test of the experimental and control groups' post-test

Group	N	M	SD	t	p
Experimental Group	23	15.2	3.70	5.600	.113
Control Group	28	13.39	2.83		

The post-test mean scores of the experimental and control groups for FMKt were found to be 15.2 and 13.39 respectively. An independent sample t-test showed there is no statistical difference ($t=5.600$, $p>0.05$) between the experimental and control groups. According to this, the pre and post-test difference averages of the experimental and control groups are not statistically different from each other. The score of the experimental group is ($M=15.0$) and the score of the control group is ($M=13.39$).

4.3 Results for Qualitative Data Based on the Activity Evaluation Form

The experimental groups were given an activity evaluation form after the treatments. Throughout the research, science lessons integrated with Scratch were given in the experimental group. At the end of the research, students' opinions about Scratch activities were collected with the activity evaluation form. Since there were 23 students in the experimental group, the activity opinion forms of 23 students were evaluated. In this research, content analysis was performed on the data obtained from the open-ended activity evaluation form. Frequency percentages of student responses were found. While the codes were being constructed it was noticed that a student's answers fit more than one code. Sample sentences from each of the study participants were attempted to be included. Examples of student responses are presented along with the numbers assigned to each student. The results obtained are given below.

Table 4.11 Students' opinions on the meaning of Scratch

Categories	Codes	Frequencies (%)	Examples of Comments
Meaning of Scratch	Game program	86.9	We can design and play fun games in Scratch. (Student 12)
	Course program	60.8	In Scratch, we learn the forces of objects in science class, so we can move objects. (Student 14)
	Fun site	43.4	We do fun lessons in Scratch. (Student 3)
	The program that makes learning easier	17.3	We learned the subject of force and motion more easily in Scratch. (Student 2)
	Project creation application	14.1	Scratch is a project creation application through games. (Student 8)

Table 4.11 Students' opinions on the meaning of Scratch (continued…)

	Coding application	14.1	Scratch is a coding tool. (Student 16)
	Learning while playing a program	14.1	We both play and design games when studying with Scratch, it's very fun, and it helps us learn while playing. (Student 21)

As seen in Table 4.11, 86.9 percent of 23 students define Scratch as a game program. Additionally, the majority of students define Scratch as a course application. Most of the students stated that they had fun lessons with Scratch. 14.1 percent of students define Scratch as Project creation, coding and learning while playing a program. Remarkable aspects of student answers are most of the students consider Scratch to be an enjoyable game program so students learn while having fun.

Table 4.12 Students' opinions on reasons for using Scratch

Categories	**Codes**	**Frequencies (%)**	**Examples of Comments**
Reasons for using Scratch	Studying	86.9	We taught with Scratch, and we had a lot of fun, I was very happy. (Student 5)
	Playing a game	65.2	Scratch allows us to create any kind of game we want and play with that game. (Student 22)
	Designing a project	30.4	I designed a project on force and motion unit, and I played with that project. (Student 19)

Table 4.12 Students' opinions on reasons for using Scratch (continued…)

	Making an object move and talk	30.4	We can make an object move and talk. (Student 18)
	Designing a game	21.7	We can create games for our classes and use this method to learn subjects. (Student 20)

When students are asked what they do with Scratch, 86.9 percent of them said they study, while 65.2 of them said they play games. Furthermore, 30.4 percent of students said they designed projects in Scratch, and 30.4 percent of students said they created speaking and moving objects. At the same time, 21.7 percent of students stated that they designed games in Scratch.

Table 4.13 Students' opinions on using Scratch in other areas of life

Categories	Codes	Frequencies (%)	Examples of Comments
Using Scratch in other areas of life	Mathematics lesson	47.8	We can create a project in mathematics class. (Student 10)
	Science lesson	26.0	I made other projects at home with Scratch on the subject we learned in science class. (Student 3)
	Computer Technologies	26.0	We can do different projects in Computer Technologies class. Student 15)

Table 4.13 Students' opinions on using Scratch in other areas of life (continued…)

	Game	17.4	I can play games on Scratch with my friends. (Student 17)
	Calculation	13.0	We can use Scratch in every aspect of our lives such as we can make calculations with the project we make. (Student 12)
	Turkish lesson	8.6	It can help me do my project homework in Turkish class. (Student 7)

When students were asked in what other areas of their lives they could use Scratch, 47.8 percent said math lessons. Additionally, 26.0 percent of students stated that they would use Scratch in science and computer classes. 17.4 percent noted they would use it in games and 13.0 percent stated they would use it for calculation. 8.6 percent of students mentioned they would use Scratch in Turkish classes.

Table 4.14 Students' opinions on sources of the project idea

Categories	Codes	Frequencies (%)	Examples of Comments
Sources of the project idea	My own choices	65.2	I designed the project I wanted, so I was very happy. (Student 17)
	Doing research	21.7	I researched other people's work in Scratch. (Student 19)

Table 4.14 Students' opinions on sources of the project idea (continued…)

	With the support of my teacher	21.7	I told my teacher about the project I wanted to do and asked for help. (Student 11)
	With the help of my friends	21.7	I watched what my friends were doing and asked them for help. (Student 18)

When students were asked how they came to the idea of the project they prepared in Scratch, 65 percent of the students answered by their own decision. 21.7 percent of the students stated that they designed a project by doing research, getting support from their friends, and getting help from their teacher.

Table 4.15 Students' opinions on solution to scratch challenges

Categories	Codes	Frequencies (%)	Examples of Comments
Solution to scratch challenges	Get support from the teacher	52.1	When my code blocks did not work as I wanted, I got support from my teacher. (Student 9)
	Researching how to solve the problem	39.1	When I had difficulty doing the project, I researched Google. (Student 3)
	Get help from friends	21.7	I got help from my friend next to me. (Student 15)

When students were asked how they overcame a challenge they experienced while working on a project, 52.1 percent of the students responded that they got assistance from their teachers in this regard. 39.1 percent of students said they researched to find a solution to this problem. 21.7 percent of the students stated that they received assistance from their friends.

Table 4.16 Students' opinions on favorite aspects of Scratch

Categories	Codes	Frequencies (%)	Examples of Comments
Favorite aspects of Scratch	Creating projects using various environments, characters, and voices	52.1	I loved creating projects using the characters I wanted, backgrounds and sounds in Scratch. (Student 18)
	Doing a wide variety of projects	26.0	It was great to produce a wide variety of projects in Scratch. (Student 23)
	Designing my project freely	21.7	One of my favorite features while doing the project is that I have the freedom to choose the background I want and the characters I want. (Student 7)
	Being funny	17.4	One of my favorite aspects of Scratch is it gives us fun. (Student 13)
	Seeing the project, I designed working	13.0	I was very happy to see that my project idea worked. (Student 4)

When students asked what aspects of Scratch they like, 52.1 percent of them mentioned the diversity of scenarios and characters they could use in their projects, 26.0 percent of students said that the ability to make the characters they create talk. Additionally, 21.7 percent of students mentioned that it gives them the freedom to design their projects as they wish. While 17.4 percent of the students stated that one of their favorite aspects of Scratch was that it was entertaining, 13.0 percent of the students stated that seeing the project they designed working was one of their favorite aspects.

Table 4.17 Students' opinions on dislikes of Scratch

Categories	Codes	Frequencies (%)	Examples of Comments
Dislikes of Scratch	Difficulty locating code blocks	26.0	I didn't like it when the blocks of code I placed didn't work the way I wanted. (Student 5)
	Being confusing	21.7	When I first started using Scratch, I had a hard time moving the code blocks around. (Student 15)

When students were asked about the features they did not like in Scratch, 26.0 percent of them stated that they had difficulty placing the code blocks and 21.7 percent of students stated that the program was confusing.

Table 4.18 Students' relationships with technology

Categories	Codes	Frequencies (%)	Examples of Comments
Students' relationships with technology	Playing a game	56.5	I like playing online games. (Student 22)

Table 4.18 Students' relationships with technology (continued…)

Categories	Codes	Frequencies (%)	Examples of Comments
Students' relationships with technology	Playing a game	56.5	I like playing online games. (Student 22)
	Studying lesson	52.1	I like to take tests online after studying. (Student 13)
	Watching video/movie	30.4	I enjoy watching online videos that I wish to. (Student 11)
	Painting	17.4	I like painting on the computer by discovering new programs. (Student 20)
	Doing research	13.0	We can do research on Google in a short time without spending hours researching in books. (Student 17)
	Online chat	4.3	I like chatting with my friends on the phone. (Student 5)

Finally, students were asked what they like to do with technology. 56 percent of the students stated that they like playing online games, and 52 percent of them said that they like studying with technology. Furthermore, 30.4 percent of the students mentioned using technology to watch movies or videos, followed by 17.4 percent of them said painting, 13.0 percent of them said they conduct research, and 4.3 percent of them commented that they chat with their friends.

5
DISCUSSION

The study aimed to evaluate the impact of integrating a block-based programming environment into science lessons for sixth-grade students to enhance their achievement and CT. One of the key assessments conducted in the research was the administration of the CTt developed by Roman-González et al. (2017) to middle school students. The test underwent validation and reliability analyses, including adaptation from English to Turkish to ensure its appropriateness for the study context. Employing a quasi-experimental research design, the study divided participants into a control group, which followed the Ministry of National Education's teaching methodology, and an experimental group that utilized Scratch programming environments. Over a six-week period in a public middle school, a classroom intervention was implemented to help students understand concepts related to force and motion, as well as CT, through the utilization of block-based programming tools. Furthermore, the research explored the perceptions and opinions of students receiving science education integrated with block-based programming regarding their experiences with Scratch. This investigation aimed to provide insights into students' attitudes towards this educational approach. Block-based programming environments are becoming more and more popular in science classrooms, which highlights the continued efforts of researchers and teachers to improve students' understanding of concepts in science and improve their CT abilities through creative and interesting teaching methods (Basu et al., 2013; Lee et al., 2020; Peel et al., 2019; Weintrop, 2019). In this section, the results of integrating block-based programming environments into education regarding students' achievement, CT, and opinions are discussed by related studies in the relevant literature.

This study shows the effect of block-based programming environments on the development of students' CT and conceptual understanding in science teaching. The use of Scratch programming environments has a fundamental effect, as seen by the notable changes in sixth-grade students' CT after the intervention. The results of the research are consistent with other studies in the area. To promote CT abilities, Cetin (2016) underlined the need of incorporating programming environments into educational settings. Cetin (2016) used a mixed-method design to discuss preservice teachers' learning and attitudes toward programming to assess the effects of Scratch-based instruction. The findings showed that preservice teachers who were taught using Scratch had a considerably greater comprehension of computational concepts than those who were not. Aytekin & Topçu, (2023) provided additional evidence for this idea by emphasizing how programming interventions improve students' CT and achievement of science topic. They designed plugged and unplugged lessons that included CT components into science lessons. The results of the experimental study showed that, in comparison to the control group, both experimental groups with plugged and unplugged courses achieved significantly higher accomplishment in CT skills and achievement of the science topic. In the current study, the academic achievement test scores of the students in the experimental group do not show a significant difference compared to the students in the control group, but the averages of the students in the experimental group are higher than those in the control group.

Post-intervention evaluation revealed that, after receiving traditional teaching, the experimental group exhibited noticeably better CT than the control group. This result aligns with Luo et al. (2020) study, which showed that after taking part in a science unit that combined CT using the Dash robot and the Blockly app, primary school students demonstrated improved CT skills. These studies demonstrate how effective it is to use CT in science classes to help students understand concepts more deeply and become more engaged. Furthermore, even though our study did not find significant differences in the experimental and control groups' achievement of force and motion, the experimental group's overall improvement indicates that block-based programming can be an effective tool for improving individual learning. This conclusion supports (Luo et al., 2020) findings that CT instruction can improve CT understanding even in low-interest topics when combined with scaffolded exercises and rapid feedback. Luo et al.'s (2020) results that CT exercises might attract students' attention and recall in scientific contexts are supported by the generally positive feedback that Scratch received from students in our study.

According to the current study's findings, sixth-grade students' CT and achievement of science topic significantly improved after the intervention, which is consistent with Aksit & Wiebe (2020) explored the effects of teaching simulation-based modeling and CT using block-based programming on seventh-grade students' CT abilities and comprehension of force and motion principles. The findings suggest that immersive experiences with programming tools enhance CT skills and achievement. The effectiveness of using Scratch programming environments in science classes is shown by the current study, which shows significant improvements in sixth-grade students' achievement of force and motion after the intervention, but there are no significant differences between the control and experimental groups. This result is consistent with Hutchins et al. (2020) investigation of the Collaborative, Computational STEM (C2STEM) setting, in which students demonstrated increased motivation and engagement in building models to comprehend physics concepts when faced with difficulties, highlighting the positive effects of practical, problem-solving experiences in STEM education.

This study's methodology and results are consistent with other studies showing that block-based programming environments can improve scientific literacy and computational abilities. Grover et al. (2015), for instance, stress that computational thinking is fundamental to scientific knowledge and inquiry and that incorporating it into science education can result in more engaging learning opportunities. Similarly, S. A. Papert (1980) suggests that programming promotes critical thinking in students, an essential component of scientific literacy. Students are introduced to critical thinking and problem-solving exercises, which are essential components of scientific literacy, by integrating programming into the science curriculum. Students' comprehension of science subjects is strengthened by these assignments, which encourage them to make hypotheses, test theories, and evaluate evidence in ways that are like the scientific process (NGSS Lead States, 2013).

Furthermore, as highlighted by (Kazakoff et al., 2013) supports the idea that teaching block-based programming languages such as Scratch to students can help them develop CT abilities. Students learn how to write as well as develop key computational concepts like abstraction, conditionals, loops, and sequencing by being engaged in a creative and interactive programming environment. In the current study, the averages of the sub-dimensions of adapted CTt by researcher which are basic directions and sequences, loops-repeat times, loops repeat until, if-simple conditional, if/else-complex conditional, while conditional, simple functions measured by the CTt were increased according to the results of the experimental group’s post-test score after the intervention. The results related to CT, are consistent with the findings of Sáez-López et al. (2016) studied the usage of a visual programming language, specifically Scratch, in the classroom by examining the attitudes and performance of fifth and sixth grade primary students. The study found that elementary school students who used Scratch improved significantly in their comprehension of computational practices, logic, and programming concepts. Additionally, students demonstrated increased enthusiasm, motivation, passion, and commitment in addition to gains in CT and computational practices.

The expanding body of research on the relationship between programming tools, CT development, and science education has also benefited from remarkable works by Hutchins et al. (2020), Aytekin & Topçu (2023), and Basu et al. (2013). The results of this study, together with earlier research, offer valuable information for educators and researchers who want to improve students' learning experiences considering the rapidly changing field of educational technology. In a biology classroom, Peel et al. (2019) study employed unplugged activities to investigate CT principles, while my research concentrated on using block-based programming environments to teach subjects in science. According to Peel et al. (2019), including CT into natural selection classes improved students' understanding of the subject. In the same way, using block-based programming tools improved students' understanding of force and motion concepts in the current study, demonstrating the value of integrating CT principles into scientific instruction to improve student learning outcomes.

The study's findings highlight the potential for developing achievement and CT abilities in sixth-grade students by incorporating block-based programming environments into science classes. According to Lee et al. (2020) and (Wing, 2006, 2008, 2014), further investigation and study are necessary to advance the field of educational technology and its effects on student learning outcomes going forward. Remarkably, the views of students who experienced scientific instruction combined with block-based programming environments specifically, Scratch was favorable. This supports the findings of earlier research, such as Lee et al. (2020) and Weintrop, (2019) which highlight the importance of students' attitudes and views of these advanced teaching tools. Teachers and researchers can investigate the advantages of integrating these technologies into curriculum design and instructional methods for better learning outcomes in STEM courses by considering the insights offered by the relevant literature. Cetin, (2016) used a mixed-method design to discuss preservice teachers' learning and attitudes toward programming to assess the effects of Scratch-based instruction. The findings showed that preservice teachers who were taught using Scratch had a considerably greater comprehension of computational concepts than those who were not also they were more enthusiastic about programming. Additionally, students in Lai & Lai (2012) study stated that they are willing to do additional computer programming assignments for future science courses. These findings are consistent with the findings of the second question of this study's activity evaluation form, asking in which areas of their life's students use Scratch. The positive perspectives that students had of Scratch in the current study are consistent with other studies, such Cetin (2016), which emphasized the benefits of Scratch-based learning. In Cetin's study, the experimental group also conveyed happiness for the lessons learned, especially about the creative outcomes (like game development) that came from their coding efforts. In this regard, the excitement expressed by some students in the experimental group when they experienced the completion of their projects emphasizes how engaging and enjoyable Scratch-based instruction is.

The study's conclusions highlight the possibility of using block-based programming environments, such as Scratch, to improve sixth-grade students' achievement and CT in science classes.

6
CONCLUSION

This study was essential in demonstrating the application of block-based programming environments in educational settings. Despite being referenced in the curriculum, block-based programming environments are not widely used in education, particularly when it comes to teaching science. The study's conclusions highlight how incorporating block-based programming environments more especially, Scratch into science classes for sixth-grade students could enhance their achievement as well as CT. Few research has been done on how CT integration should occur in the setting of multidisciplinary studies, despite the abundance of studies in the literature on the value of CT and its integration in the classroom. The study showed that sixth-grade students' CT significantly improved after the intervention, demonstrating the potential of block-based programming to improve children's CT. To test students' CT, the researcher adapted the CTt to Turkish. Furthermore, the comparison of the experimental and control groups revealed significant differences in CT, supporting the idea that students' practical experience with Scratch programming environments might result in observable improvements in CT. This emphasizes how important it is to use technology-enhanced scientific instruction to improve CT. The study revealed that after the intervention, students' achievement of force and motion had significantly improved. Although the control and experimental groups did not differ significantly in this regard, the overall improvement in students' understanding of scientific ideas highlights the value of using block-based programming tools to support learning objectives. Additionally, students' generally positive opinions about Scratch and its incorporation into science classes show how engaged and satisfied they are with block-based programming environments.

Overall, the results of this study have several valuable implications for science educators. First, schools should explore incorporating block-based programming environments like Scratch into the wider curriculum, especially in science instruction, to further improve students' CT. Second, opportunities for professional development should be made available to educators so they may successfully integrate Scratch and other block-based programming tools into their lesson plans.

Appropriate training will provide educators the abilities needed to teach classes that are both interesting and effective. Third, future studies need to examine the long-term effects of Scratch on other learning domains, such as creativity, critical thinking, and problem-solving abilities in students. To investigate if the same improvements are shown, it would also be beneficial to expand this research to other topics and grade levels.

REFERENCES

Adler, R. F., & Kim, H. (2018). Enhancing future K-8 teachers' computational thinking skills through modeling and simulations. *Education and Information Technologies*, *23*, 1501–1514.

Aksit, O. (2018). *Enhancing science learning through computational thinking and modeling in middle school classrooms: A mixed methods study*. North Carolina State University.

Aksit, O., & Wiebe, E. N. (2020). Exploring force and motion concepts in middle grades using computational modeling: A classroom intervention study. *Journal of Science Education and Technology*, *29*(1), 65–82.

Angevine, C., Cator, K., Roschelle, J., Thomas, S. A., Waite, C., & Weisgrau, J. (2017). *Computational thinking for a computational world*.

Ardito, G., Mosley, P., & Scollins, L. (2014). We, robot: Using robotics to promote collaborative and mathematics learning in a middle school classroom. *Middle Grades Research Journal*, *9*(3).

Aydin, A., Aslan, A., & Aydinbelge, B. (2022). *Ortaokul ve İmam Hatip Ortaokulu Fen Bilimleri 6. Sınıf Ders Kitabi* (S. Yilmaz, Ed.).

Aytekin, A., & Topçu, M. S. (2023). The effect of integrating computational thinking (CT) components into science teaching on 6th grade students' learning of the circulatory system concepts and CT skills. *Education and Information Technologies*, 1–32.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, *38*(6), 20–23.

Basu, S., Dickes, A., Kinnebrew, J. S., Sengupta, P., & Biswas, G. (2013). CTSiM: A computational thinking environment for learning science through simulation and modeling. *International Conference on Computer Supported Education*, *2*, 369–378.

Bian, X., Qin, B., Xin, X., Li, J., Su, X., & Wang, Y. (2022). Handwritten mathematical expression recognition via attention aggregation based bidirectional mutual learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(1), 113–121.

Boticki, I., Pivalica, D., & Seow, P. (2018). The use of computational thinking concepts in early primary school. *Science*, *2*, 1.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada*, *1*, 25.

Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2020). Bilimsel araştırma yöntemleri (28. Baskı). *Pegem Akademi*.

Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with Scratch: An experiment with 6th grade students. *Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, September 15-18, 2015, Proceedings 10*, 17–27.

Cetin, I. (2016). Preservice teachers' introduction to computing: exploring utilization of scratch. *Journal of Educational Computing Research*, *54*(7), 997–1021.

Chuechote, S., Nokkaew, A., Phongsasıthorn, A., & Laosınchaı, P. (2020). A neo-piagetian analysis of algorithmic thinking development through the "sorted" digital game. *Contemporary Educational Technology*, *12*(1), 1–15.

Çiftçi, A., & Topçu, M. S. (2023). Improving early childhood pre-service teachers' computational thinking teaching self-efficacy beliefs in a STEM course. *Research in Science & Technological Education*, *41*(4), 1215–1241.

Çimentepe, E. (2019). *STEM etkinliklerinin akademik başarı, bilimsel süreç becerileri ve bilgisayarca düşünme becerilerine etkisi*. Eğitim Bilimleri Enstitüsü.

Creswell-John, W., & Creswell-David, J. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches.* Sage Publications.

Dutta, S., & Mathur, R. (2011). Computer programming-A building block of STEM. *2011 Integrated STEM Education Conference (ISEC)*, 2B – 1.

Erümit, A. K. (2020). Effects of different teaching approaches on programming skills. *Education and Information Technologies*, *25*(2), 1013–1037.

Fosnot, C. T., & Perry, R. S. (1996). Constructivism: A psychological theory of learning. *Constructivism: Theory, Perspectives, and Practice*, *2*(1), 8–33.

Fraenkel-Jack, R., Wallen-Norman, E., & Hyun-Halen, H. (1993). *How to design and evaluate research in education. NY: McGraw-Hill Higher Education* (10th ed.).

Fraillon, J., Ainley, J., Schulz, W., Duckworth, D., & Friedman, T. (2019). *IEA international computer and information literacy study 2018 assessment framework*. Springer Nature.

Geng, H. (2021). Redefining the role of teachers in developing critical thinking within the digital era. *2021 International Conference on Modern Educational Technology and Social Sciences (ICMETSS 2021)*, 18–21.

Goldenberg, E. P., & Carter, C. J. (2021). Programming as a language for young children to express and explore mathematics in school. *British Journal of Educational Technology*, *52*(3), 969–985.

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

Güven, I., & Gulbahar, Y. (2020). Integrating computational thinking into social studies. *The Social Studies*, *111*(5), 234–248.

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. (2010). Multivariate Data Analysis: Pearson Education. *Upper Saddle River, New Jersey*.

Halverson, E. R., & Sheridan, K. (2014). Arts education and the learning sciences. *The Cambridge Handbook of the Learning Sciences*, 626–648.

Higgins, S. (2014). Critical thinking for 21st-century education: A cyber-tooth curriculum? *Prospects*, *44*(4), 559–574.

Hsu, T.-C., & Liang, Y.-S. (2021). Simultaneously improving computational thinking and foreign language learning: Interdisciplinary media with plugged and unplugged approaches. *Journal of Educational Computing Research*, *59*(6), 1184–1207.

Hurt, T., Greenwald, E., Allan, S., Cannady, M. A., Krakowski, A., Brodsky, L., Collins, M. A., Montgomery, R., & Dorph, R. (2023). The computational thinking for science (CT-S) framework: Operationalizing CT-S for K–12 science education researchers and educators. *International Journal of STEM Education*, *10*(1), 1.

Hutchins, N. M., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., Blair, K. P., Chin, D., Conlin, L., & Basu, S. (2020). C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, *29*, 83–100.

James-McMillan, H., & Schumacher, S. (2010). *Research in education: Evidence-based inquiry*. Pearson Higher Ed.

Kajander, A. (2018). Learning algebra with models and reasoning. *Teaching and Learning Secondary School Mathematics: Canadian Perspectives in an International Context*, 561–569.

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*, 245–255.

Kotsopoulos, D., Floyd, L., Dickson, B. A., Nelson, V., & Makosz, S. (2022). Noticing and naming computational thinking during play. *Early Childhood Education Journal*, *50*(4), 699–708.

Lai, C.-S., & Lai, M.-H. (2012). Using computer programming to enhance science learning for 5th graders in Taipei. *2012 International Symposium on Computer, Consumer and Control*, 146–148.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, *29*, 1–8.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. ACM Inroads, 2 (1), 32–37. *DOI: Https://Doi. Org/10.1145/1929887.1929902*.

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. In *Journal for STEM Education Research* (Vol. 3, pp. 1–18). Springer.

Luo, F., Antonenko, P. D., & Davis, E. C. (2020). Exploring the evolution of two girls' conceptions and practices in computational thinking in science. *Computers & Education*, *146*, 103759.

Luxton-Reilly, A., Becker, B. A., Cao, Y., McDermott, R., Mirolo, C., Mühling, A., Petersen, A., Sanders, K., Simon, & Whalley, J. (2018). Developing assessments to determine mastery of programming fundamentals. *Proceedings of the 2017 ITiCSE Conference on Working Group Reports*, 47–69.

Marcum-Dietrich, N. I., Bruozas, M., Becker-Klein, R., Hoffman, E., & Staudt, C. (2024). Precipitating Change: Integrating Computational Thinking in Middle School Weather Forecasting. *Journal of Science Education and Technology*. https://doi.org/10.1007/s10956-024-10095-y

Minda, A. A., Gillich, N., Chioncel, C. P., & IosifPraisach, Z. (2015). Enhancing Mathematical Skills By The Use Of Virtual Instruments. *Procedia-Social and Behavioral Sciences*, *191*, 996–1001.

Ministry of National Education. (2018). *Fen Bilimleri Dersi Öğretim Programi (İlkokul ve Ortaokul 3, 4, 5, 6, 7 ve 8. Sınıflar)*.

Moreno-León, J., Román-González, M., & Robles, G. (2018). On computational thinking as a universal skill: A review of the latest research on this ability. *2018 IEEE Global Engineering Education Conference (EDUCON)*, 1684–1689.

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

Nugent, G., Barker, B., Grandgenett, N., & Welch, G. (2016). Robotics camps, clubs, and competitions: Results from a US robotics project. *Robotics and Autonomous Systems*, *75*, 686–691.

Nunnally, J., & Bernstein, I. (1994). *Psychometric Theory 3rd edition (MacGraw-Hill, New York)*.

Nurhayati, E., Rizaldi, D. R., & Fatimah, Z. (2020). The Correlation of Digital Literation and STEM Integration to Improve Indonesian Students' Skills in 21st Century. *Online Submission*, *1*(2), 73–80.

Ogegbo, A. A., & Ramnarain, U. (2022). A systematic review of computational thinking in science classrooms. *Studies in Science Education*, *58*(2), 203–230.

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.

Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, *99*, 415–427.

Papert, S. (1993). The children's machine: Rethinking school in the age of the computer. *New York*.

Papert, S. A. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic books.

Peel, A., Sadler, T. D., & Friedrichsen, P. (2019). Learning natural selection through computational thinking: Unplugged design of algorithmic explanations. *Journal of Research in Science Teaching*, *56*(7), 983–1007.

Pietros, J., Sweetman, S., & Shim, M. (2022). How is Computational Thinking Defined in Elementary Science? *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2*, 1112.

Repenning, A., Ioannidou, A., & Zola, J. (2000). AgentSheets: End-user programmable simulations. *Journal of Artificial Societies and Social Simulation*, *3*(3), 351–358.

Resnick, M., & Siegel, D. (2007). *Scratch*.

Roberts, D. A. (2013). Scientific literacy/science literacy. In *Handbook of research on science education* (pp. 729–780). Routledge.

Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, *28*(3), 316–327.

Roman-Gonzales, M. (2015). Computational Thinking Test: Design Guidelines And Content Validation. *In EDULEARN15 Proceedings*, 2436–2644.

Roman-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, *97*, 129–141.

Sands, P. (2019). Addressing cognitive load in the computer science classroom. *Acm Inroads*, *10*(1), 44–51.

Scherer, R. (2016). Learning from the past–the need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in Psychology*, *7*, 223017.

Selby, C., & Woollard, J. (2013). *Computational thinking: the developing definition*.

States, N. L. (2013). *Next generation science standards: For states, by states*. National Academies Press.

Tabach, M., Hershkowitz, R., & Arcavi, A. (2008). Learning beginning algebra with spreadsheets in a computer intensive environment. *The Journal of Mathematical Behavior*, *27*(1), 48–63.

Tsarava, K., Moeller, K., Román-González, M., Golle, J., Leifheit, L., Butz, M. V, & Ninaus, M. (2022). A cognitive definition of computational thinking in primary education. *Computers & Education*, *179*, 104425.

Tucker-Raymond, E., Puttick, G., Cassidy, M., Harteveld, C., & Troiano, G. M. (2019). "I Broke Your Game!": critique among middle schoolers designing computer games about climate change. *International Journal of STEM Education*, *6*, 1–16.

Van Aert, R. C. M., Wicherts, J. M., & van Assen, M. A. L. M. (2016). Conducting meta-analyses based on p values: Reservations and recommendations for applying p-uniform and p-curve. *Perspectives on Psychological Science*, *11*(5), 713–729.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*, 715–728.

Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM*, *62*(8), 22–25.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*, 127–147.

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM*, *57*(8), 24–28.

Wilson, C. (2013). What's Up Next for Code. org? *Computer*, *46*(8), 95–97.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

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

Wing, J. M. (2014). Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing*, *2014*, 26.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*, 565–568.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, *14*(1), 1–16.

Yıldırım, A., & Şimşek, H. (2005). Sosyal bilimlerde nitel araştırma yöntemleri.(5. baskı) Ankara: Seçkin Yayıncılık. *Center for Values Education*.

Yusoff, K. M., Ashaari, N. S., Wook, T. S. M. T., & Ali, N. M. (2020). Analysis on the requirements of computational thinking skills to overcome the difficulties in learning programming. *International Journal of Advanced Computer Science and Applications*, *11*(3).

Zendler, A., & Spannagel, C. (2008). Empirical foundation of central concepts for computer science education. *Journal on Educational Resources in Computing (JERIC)*, *8*(2), 1–15.

Zhong, B., & Xia, L. (2020). A systematic review on exploring the potential of educational robotics in mathematics education. *International Journal of Science and Mathematics Education*, *18*(1), 79–101.

A
FORCE AND MOTION KNOWLEDGE TEST (FMKt)

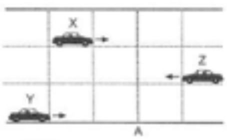
. **Akademik Başarı Testi**

Adı - Soyadı:

Sınıfı : 6 / **Numara:**

Okulu :

1.



Şekildeki doğrusal yolda, sabit süratlerle belirtilen yönlerde hareket eden X,Y ve Z araçları t=0 anında şekilde belirtilen konumlardadır. Araçlar t=5 saniye anında A çizgisinden geçtiklerine göre, süratlerinin büyüklükleri arasındaki ilişki hangi seçenekte doğru verilmiştir?

A) X > Y > Z **B)** Z > Y > X

C) Y > X = Z **D)** Y > X > Z

2. Aşağıdaki varlıklardan hangisi **dengelenmemiş** kuvvetlerin etkisindedir?

A) Ağaçta duran elma

B) Sabit süratle ilerleyen araba

C) Yokuştan aşağı hızlanan bisiklet

D) Duvardaki saat

3. Bir cisme kuzey yönünde 5N, doğu yönünde 6N, güney yönünde 5N ve batı yönünde 8N 'lik üç kuvvet aynı anda etki ediyor. Buna göre; cisme etki eden bileşke kuvvet hangi yönde ve kaç N olur?

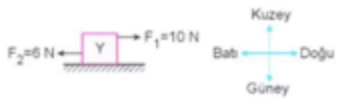
A) Batı yönünde 2N

B) Doğu yönünde 2N

C) Kuzey yönünde 14N

D) Güney yönünde 11N

4.

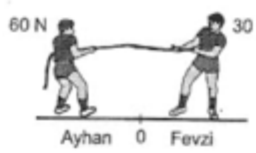


F1 ve F2 kuvvetlerinin etkisindeki Y cismine uygulanacak dengeleyici kuvvet hangi yönde kaç Newton olmalıdır?

A) Doğu yönünde 4 N **B)** Batı yönünde 16 N

C) Doğu yönünde 16 N **D)** Batı yönünde 4 N

5.



Şekilde halat çekme yarışı yapan Ayhan ve Fevzi'nin dengede kalabilmesi için hangisinin çekme yönünde kaç N' luk kuvvet eklenmelidir?

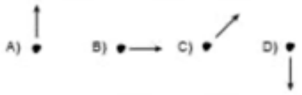
A) Ayhan'a 30 **B)** Fevzi'ye 30

C) Ayhan'a 60 **D)** Fevzi'ye 60

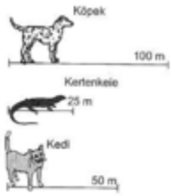
6.

Aşağıdaki şekilde bir öğrenci eline aldığı bir taşı eğik olarak fırlatıyor.

Taş en yüksek noktaya ulaştığı anda taşa etkiyen bileşke kuvvetin yönü hangi seçenekte doğru verilmiştir?



7.



Şekildeki hayvanlar aynı anda koşuya başlayıp, Aynı anda yollarını tamamlıyor. Buna göre hayvanların hızlarının büyükten küçüğe doğru sıralanışı hangisinde verilmiştir?

A) Köpek, kedi, kertenkele

B) Kertenkele, köpek, kedi

C) Köpek, kertenkele, kedi

D) Kedi, kertenkele, köpek

8. Aşağıdakilerden hangisi kuvvet ile ilgili doğru bir ifadedir?

A) Sabit sürat, sabit kuvvet gerektirir.

B) Cisme etkiyen bileşke kuvvet sıfırsa, o cismin hareketi gözlenmez.

C) Kuvvet sadece canlılar tarafından uygulanır.

D) Cismin hareket yönünü, üzerine etkiyen bileşke kuvvetin yönü belirler.

9. Bir cisme uygulanan aynı doğrultudaki üç kuvvetin büyüklükleri sırasıyla 2 N, 7 N ve 9 N'dur. Buna göre, cisme etki eden bileşke kuvvetin büyüklüğü aşağıdakilerden hangisi **olamaz?**

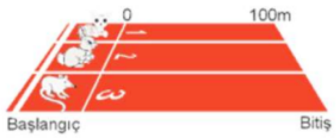
A) 0 **B)** 8 N **C)** 14 **D)** 18 N

10.

Yukarıda verilen dinamometre ile hangi ağırlıkta olan bir cisim tartılamaz?

A) 60 N **B)** 50 N **C)** 30 N **D)** 20 N

11. Üç farklı hayvan şekildeki koşu parkurunda aynı anda harekete başlıyorlar.



Bu hayvanların bitiş çizgisine ulaşma süreleri arasında Kedi > Tavşan > Fare ilişkisi vardır. Buna göre,

I. Tavşan, kediden hızlıdır.

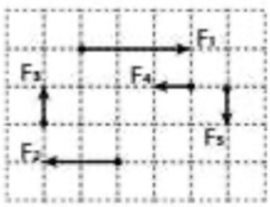
II. En süratli hayvan, faredir.

III. Kedi, tavşan ve fare eşit zamanda, eşit miktarda yol almışlardır.

A) Yalnız I **B)** Yalnız II

C) I ve II **D)** I, II ve III

12.



Verilen kuvvetlerle ilgili hangi ifade **yanlıştır**? (Bölmeler arası mesafe eşittir)

A) F3 ve F5 eşit büyüklüktedir

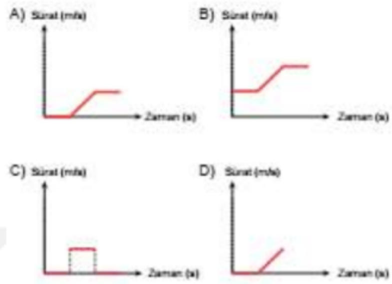
B) F2, F4'ün iki katı büyüklüktedir.

C) F1 ve F4 ters yöndedir.

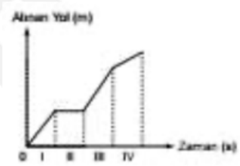
D) F1 ve F2 farklı doğrultulardadır.

13. Bir cisim sürtünmesiz bir ortamda sabit süratle hareket ederken süratli zamanla düzgün olarak artmaya başlıyor. Sürat artışından sonra yoluna tekrar sabit süratle devam ediyor.

Bu cismin sürat – zaman grafiği aşağıdakilerden hangisi olabilir?



14.



Yukarıda alınan yol – zaman grafiği verilen araç hangi zaman aralığında **vol almamıştır?**

A) I. **B)** II. **C)** III. **D)** IV.

15. Kuvvet ile ilgili aşağıda verilenlerden hangisi doğru değildir?

A) Cisme itme veya çekme şeklinde uygulanabilir.

B) Cismin şeklini değiştirir.

C) Yalnızca temas ile etkisini gösterir.

D) Cismin yönünü değiştirir.

16.

K cismine F1 ve F2 kuvvetleri uygulanıyor. F1 kuvveti ve bileşke kuvvet şekilde verilmiştir.

Buna göre K cismine uygulanan F2 kuvveti, hangi yönde kaç N' dur?

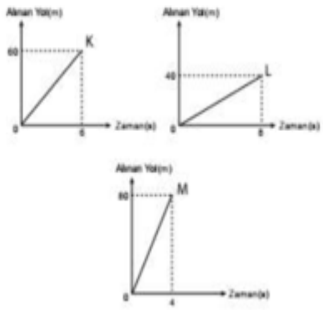
A) Doğu yönünde 1 N

B) Batı yönünde 1 N

C) Doğu yönünde 7 N

D) Batı yönünde 7 N

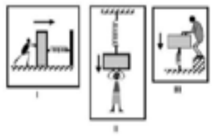
17. K, L ve M araçlarına ait yol – zaman grafikleri aşağıdaki gibidir.



Buna göre bu araçların ortalama süratleri hangisinkde doğru olarak verilmiştir?

	K	L	M
A)	10	5	20
B)	20	10	5
C)	10	20	5
D)	20	5	10

18. Üç öğrenci I, II ve III'teki yaylara oklarla gösterilen yönlerdeki kuvvetleri uyguluyorlar.



Yayların bu kişilere uyguladıkları kuvvetlerin yönleri hangi seçenekte doğru olarak verilmiştir?

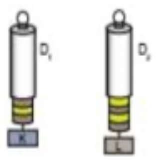
A) 

B) 

C) 

D) 

19. Şekildeki D1 ve D2 dinamometreleri özdeştir.



D1 dinamometresinde asılı olan K cisminin ağırlığı 15 N ölçüldüğüne göre, D2 dinamometresindeki L cisminin ağırlığı kaç N ölçülmüştür?

A) 15 N **B)** 20 N **C)** 25 N **D)** 30N

20. Duru ve Burak iki kardeştir. Burak kapalı haldeki odanın kapısını açmak için kolu hareket ettiriyor ve kapıyı ittiriyor. Duru ise açık olan kapıyı iterek kapatıyor. Buna göre Duru ve Burak ile ilgili;

I. Burak birden fazla kuvvet uygulamıştır.

II. Duru bir tane ve tek yönde kuvvet uygulamıştır.

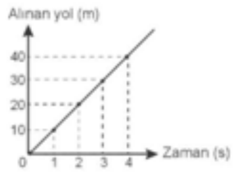
III. Duru ve Burak'ın uyguladığı kuvvetlerin yönü birbirinden farklıdır.

İfadelerinden hangileri söylenebilir?

A) I ve II **B)** I ve III

C) II ve III **D)** I, II ve III

21. Bir otomobile ait alınan yol – sürat grafiği verilmiştir.



Buna göre, bu hareketliye ait sürat – zaman çizelgesi hangisinde doğru verilmiştir?

A)

Zaman (s)	0	1	2	3	4
Sürat (m/s)	0	10	10	10	10

B)

Zaman (s)	0	1	2	3	4
Sürat (m/s)	0	10	20	30	40

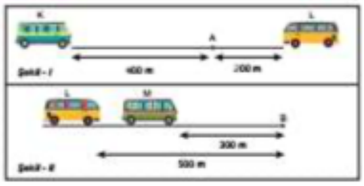
C)

Zaman (s)	0	1	2	3	4
Sürat (m/s)	0	40	30	20	10

D)

Zaman (s)	0	1	2	3	4
Sürat (m/s)	10	10	10	10	10

22. Sabit süratle birbirlerine doğru hareket eden K ve L otobüsleri Şekil I'de gösterilen A noktasında karşılaşıyorlar.



Sabit süratle aynı yöne doğru hareket eden L ve M otobüsleri ise Şekil II'de gösterilen B noktasında yan yana geliyorlar. Buna göre otobüslerin süratlerinin karşılaştırılması hangi seçenekte doğru olarak verilmiştir.

A) M > L > K B) L > K > M

C) K > L > M D) L > M > K

23. Sürtünmesiz yatay düzlemde duran piyanoya etki eden kuvvetler şekilde gösterilmiştir.

Buna göre bu kuvvetlerin;

I. Büyüklükleri,

II. Doğrultuları,

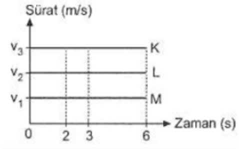
III. Yönleri,

Özelliklerinden hangileri **kesinlikle** aynıdır?

A) Yalnız II. **B)** Yalnız III.

C) I ve II. **D)** II ve III.

24.



Yukarıda K, L ve M araçlarının sürat – zaman grafikleri verilmiştir. Bu araçlardan,

- K' nın (0 – 2) saniye zaman aralığında aldığı yol X1'dir.

- L 'nin (0 – 3) saniye zaman aralığında aldığı yol X2 'dir.

- M 'nin (0 – 6) saniye zaman aralığında aldığı yol X3 'tür.

Bu araçların aldıkları yollar arasındaki ilişki X1 = X2 = X3 şeklinde olduğuna göre, araçların hızları V1, V2 ve V3 sırasıyla hangi seçenekte belirtilenler olabilir?

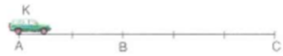
A) 30 m / s – 20 m / s – 10 m / s

B) 20 m / s – 20 m / s – 20 m / s

C) 15 m / s – 30 m / s – 40 m / s

D) 10 m / s – 20 m / s – 30 m / s

25.



Eşit bölmeli A – B – C yolunda hareket eden K aracı A – B yolunu 2t saniyede B – C yolunu 3t saniyede almıştır.

Buna göre, aşağıdaki ifadelerden hangisi doğrudur?

A) K aracının A – B yolundaki süratinin, B – C yolundaki süratine oranı 3 / 4'tür.

B) K aracının A – B yolundaki sürati, B – C yolundaki süratine eşittir.

C) K aracının A – B yolundaki süratinin, B – C yolundaki süratine oranı 4 / 3'tür.

D) K aracının A – B yolundaki sürati, B – C yolundaki süratinden büyüktür.

B
COMPUTATIONAL THINKING TEST (CTt)

BİLGİ İŞLEMSEL DÜŞÜNME TESTİ (BİDt)

Adı Soyadı:

Sınıf Düzeyi:

AÇIKLAMALAR

Test, her biri 4 sorudan oluşan 7 sayfaya bölünmüş 25 sorudan oluşmaktadır.

Tüm sorularda sadece biri doğru olan 4 cevap seçeneği (A, B, C veya D) vardır.

Testin başlangıcından itibaren 45 dakikanız vardır. Tüm soruları cevaplamanız şart değildir.

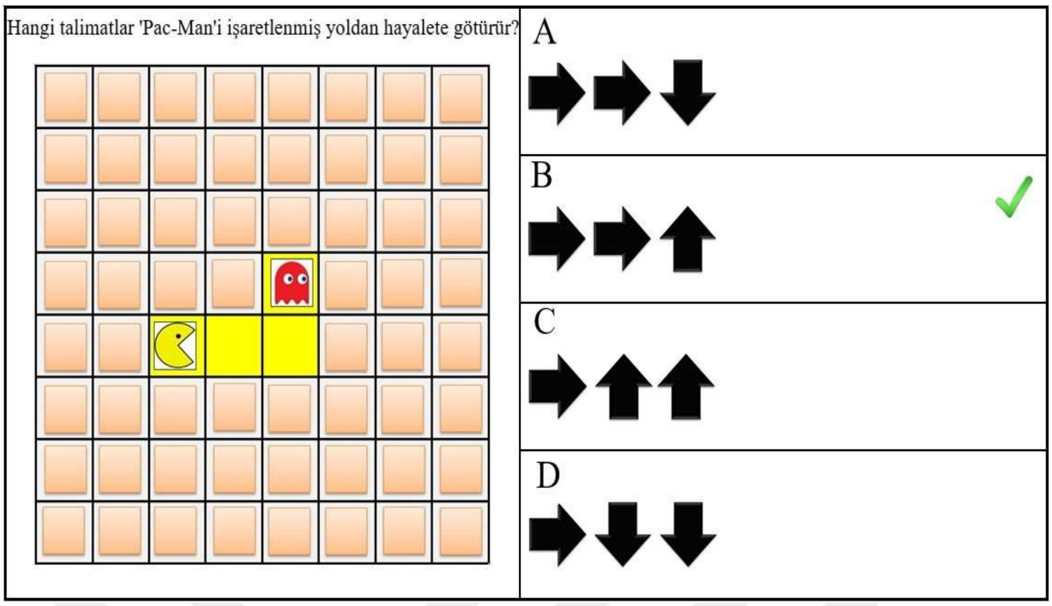
Teste başlamadan önce, karşınıza çıkacak soru türlerine ve ortaya çıkacak karakterlere aşina olmanız için üç örnek göreceksiniz.

İYİ ŞANSLAR!

1. Örnek

İlk örnekte, size hangi talimatların 'Pac-Man'i işaretlenmiş yoldan hayalete götürdüğü sorulmuştur. Yani, 'Pac-Man'i TAM OLARAK hayaletin bulunduğu kareye götürmek için (üzerinden gitmeden veya durmadan) ve sarı ile işaretlenmiş yolu kesinlikle takip etmek şartıyla (turuncu karelerle temsil edilen duvarlara dokunmadan). Bu örnekte doğru cevap B'dir.

1. Örnek: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldan hayalete götürür

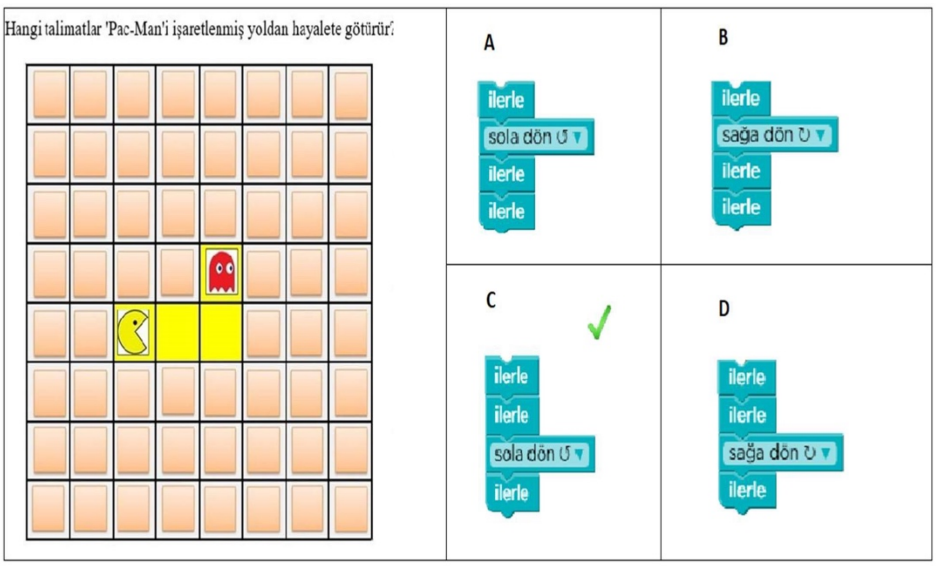


2. Örnek

İkinci örnekte, 'Pac-Man'i işaretlenmiş yoldan hayalete hangi talimatların götürdüğü tekrar sorulur. Ancak, bu soruda seçenekler oklardan ziyade talimat blokları grupları olarak sunulmaktadır. Soruda 'Pac-Man'i TAM OLARAK hayaletin bulunduğu kareye götürmenizi (üzerinden gitmeden veya durmadan) ve sarı ile işaretlenmiş yolu kesinlikle takip etmeniz gerektiğini unutmayın (turuncu karelerle temsil edilen duvarlara dokunmadan).

Bu örnekte doğru cevap C'dir.

2. Örnek: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldan hayalete götürür?

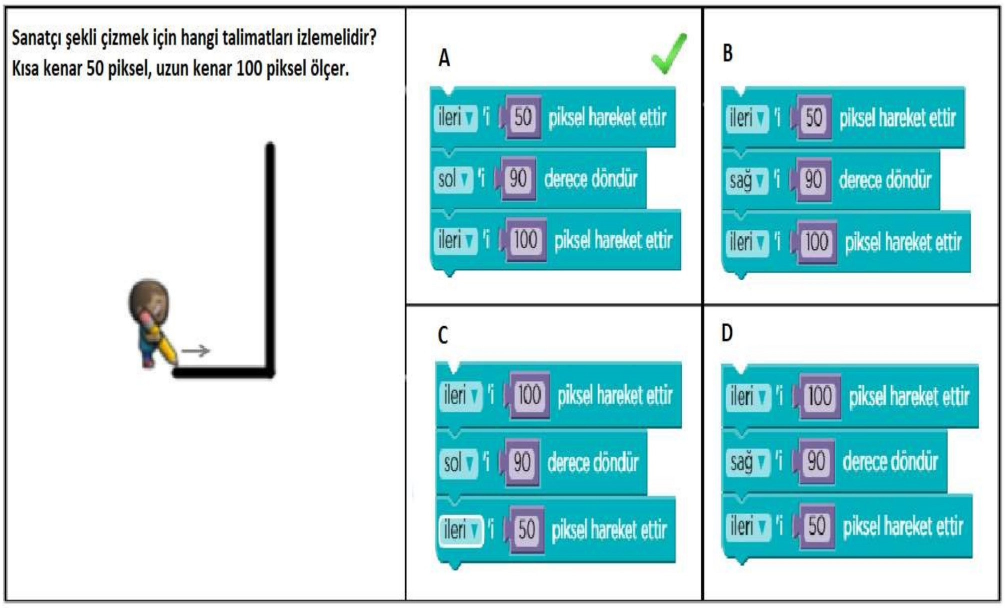


3. ÖRNEK

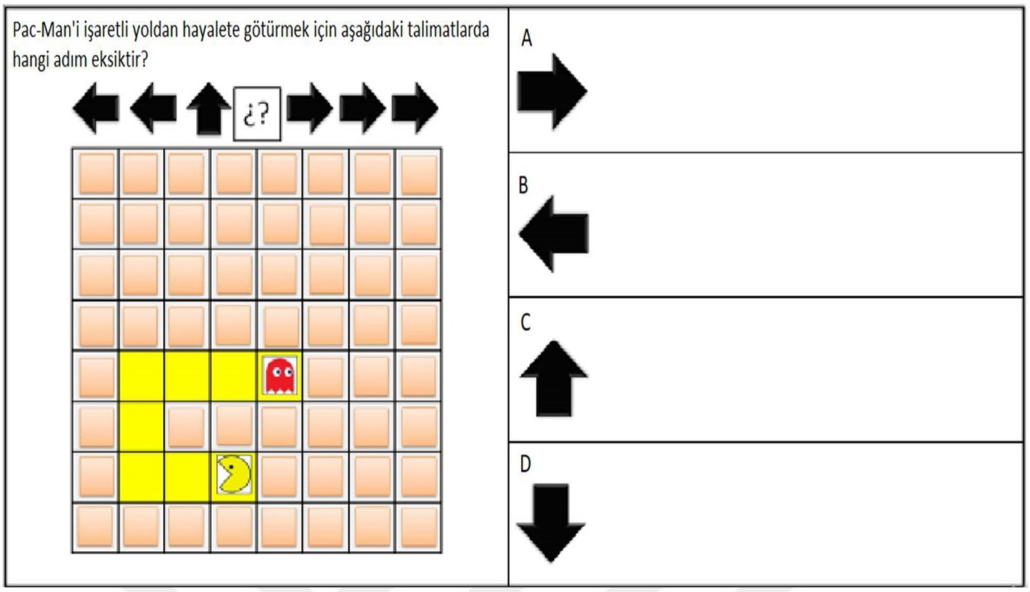
Üçüncü örnekte, sanatçının ekrandaki şekli çizmek için hangi talimatları izlemesi gerektiği sorulur. Yani, şekli çizmek için kalemi nasıl hareket ettirirsiniz? HAREKET ETTİR komutu çizim sırasında kalemi iterken, ZIPLA komutu sanatçının çizim yapmadan başka bir yere atlamasını sağlar. Gri ok, kalemin ilk hareketinin yönünü gösterir.

Bu örnekte doğru cevap A'dır.

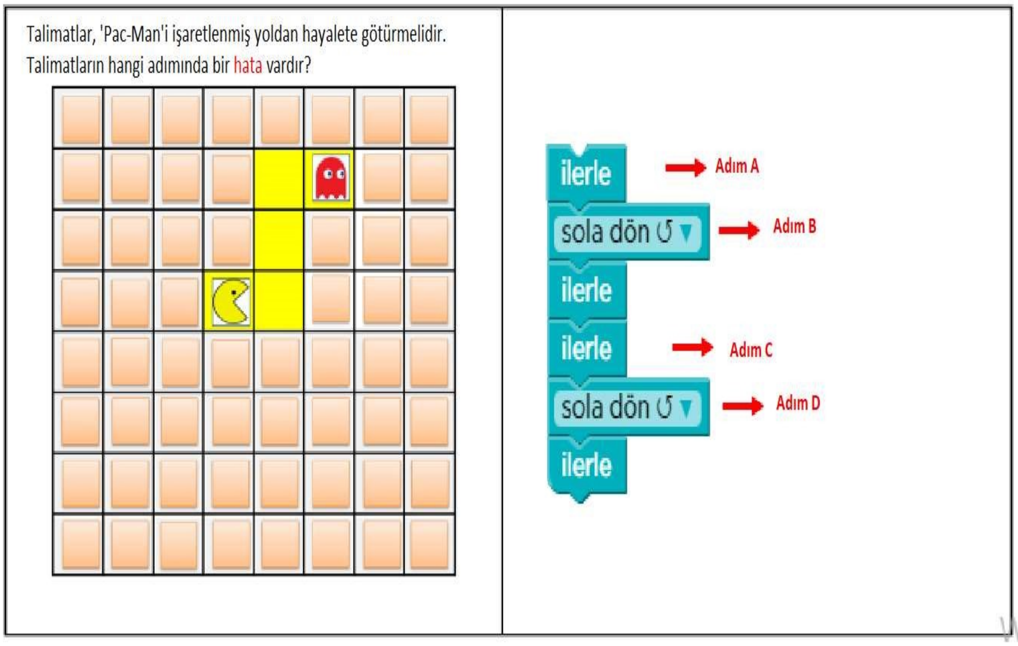
3. ÖRNEK : Sanatçı şekli çizmek için hangi talimatları izlemelidir? Kısa kenar 50 piksel, uzun kenar 100 piksel ölçer.



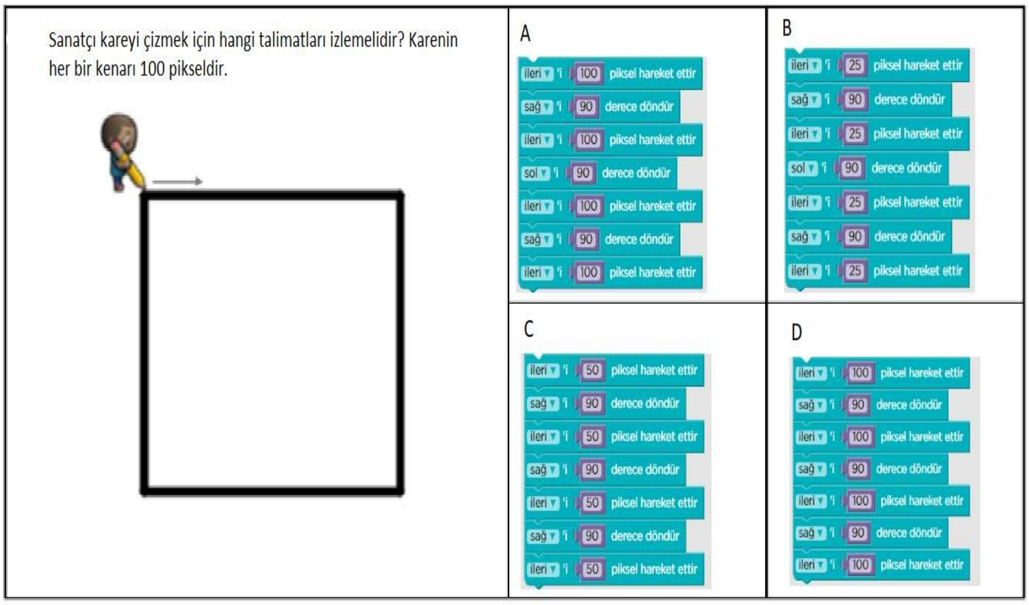
1. **Soru:** Soru: Pac-Man'i işaretli yoldan hayalete götürmek için aşağıdaki talimatlarda hangi adım eksiktir?



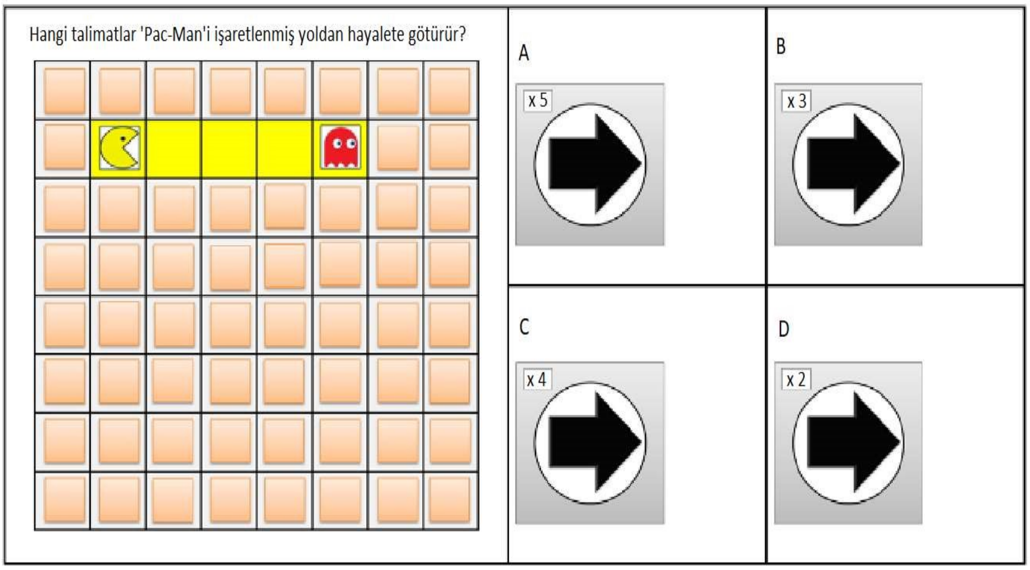
2. **Soru:** Talimatlar, 'Pac-Man'i işaretlenmiş yoldan hayalete götürmelidir. Talimatların hangi adımında bir hata vardır?



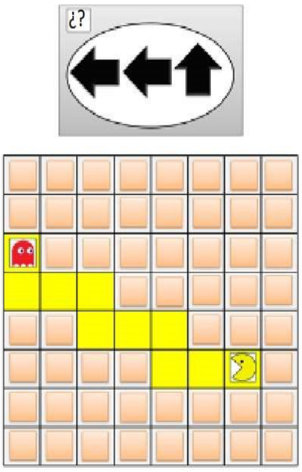
3. Soru: Sanatçı kareyi çizmek için hangi talimatları izlemelidir? Karenin her bir kenarı 100 pikseldir.



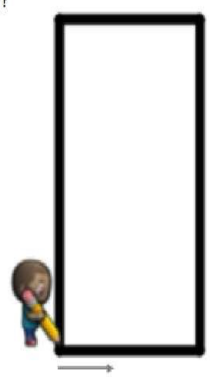
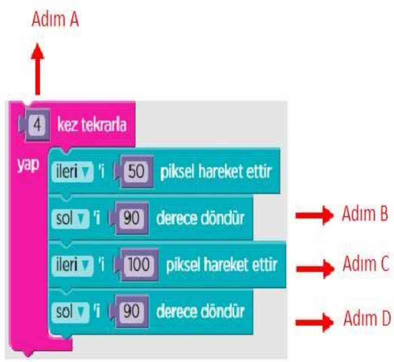
4. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldan hayalete götürür?



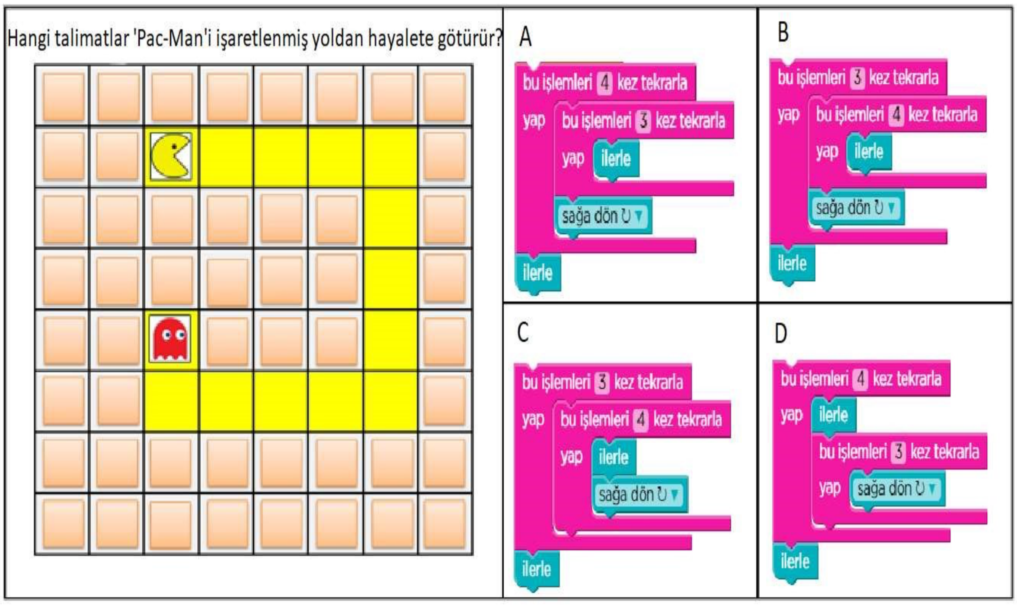
5. Soru: Pac-Man'i işaretlenmiş yoldan hayalete götürmek için dizi kaç kez tekrarlanmalıdır?

'Pac-Man'i işaretlenmiş yoldan hayalete götürmek için dizi kaç kez tekrarlanmalıdır?  	A x 2
	B x 1
	C x 4
	D x 3

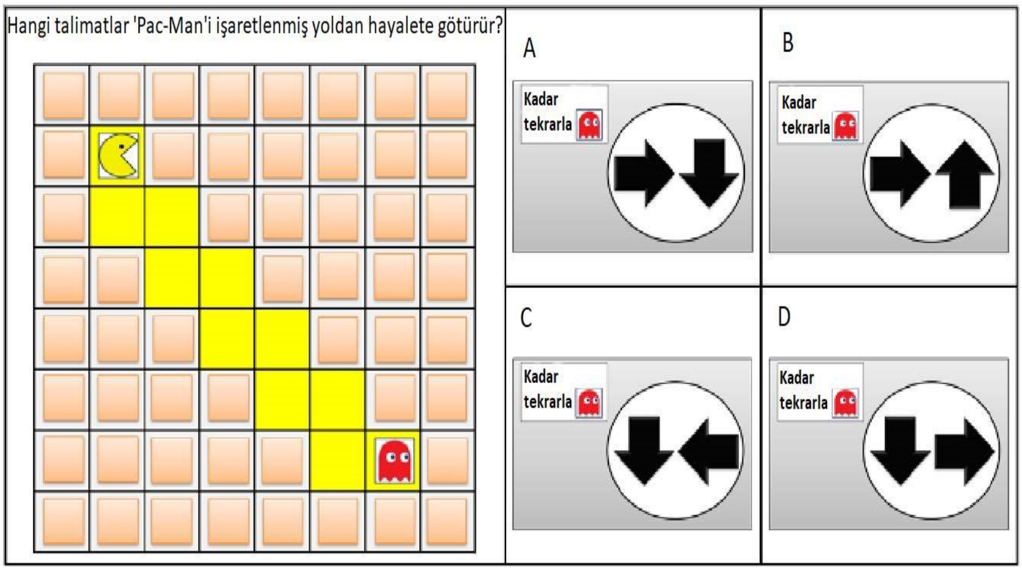
6. Soru: Talimatlar, sanatçının aşağıdaki dikdörtgeni bir kez çizmesini sağlamalıdır (50 piksel genişliğinde ve 100 piksel yüksekliğinde). Talimatların hangi adımında bir hata vardır?

Talimatlar, sanatçının aşağıdaki dikdörtgeni **bir kez** çizmesini sağlamalıdır (50 piksel genişliğinde ve 100 piksel yüksekliğinde). Talimatların hangi adımında bir hata vardır? 	

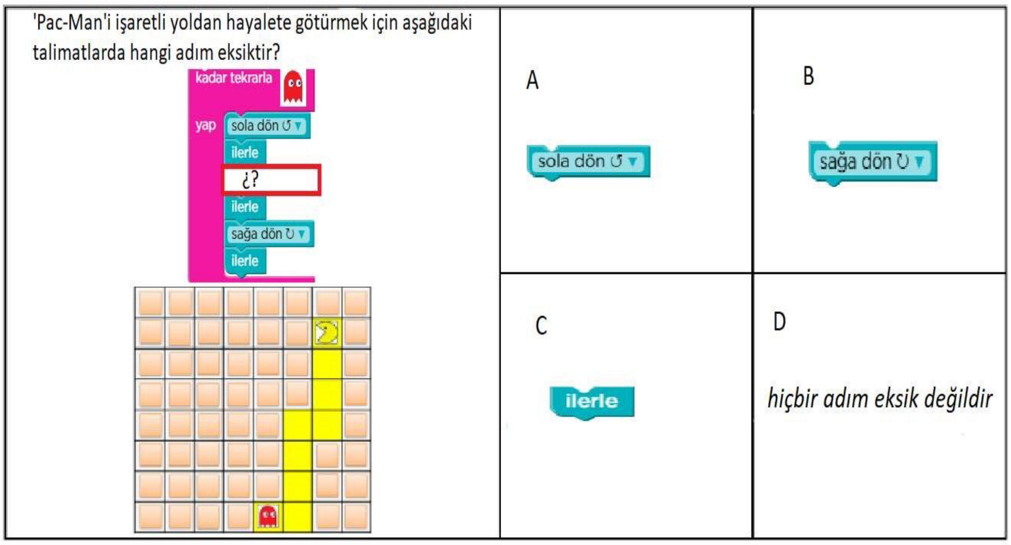
7. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldan hayalete götürür?



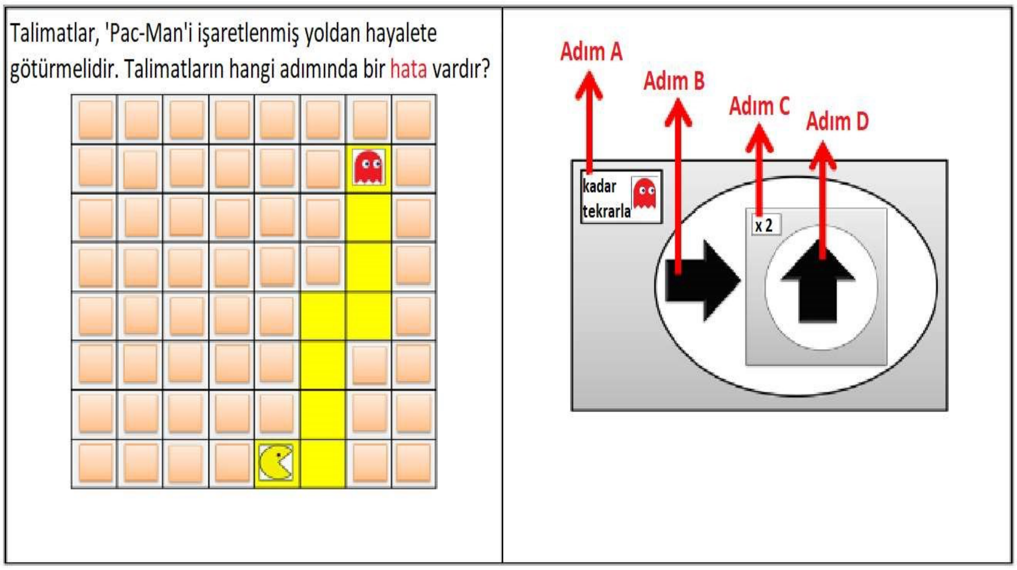
8. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldan hayalete götürür?



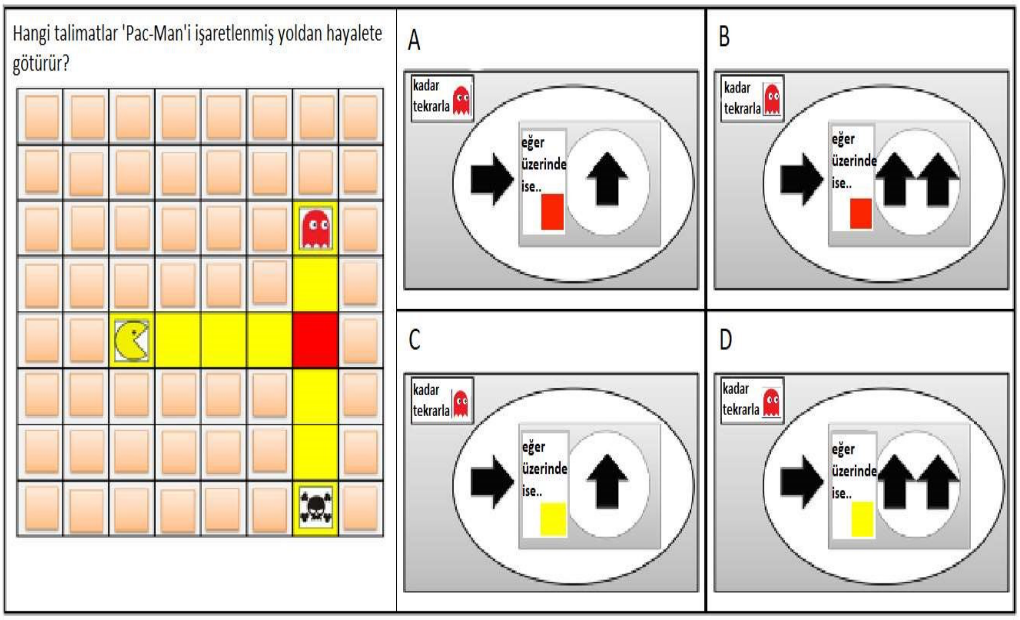
9. Soru: 'Pac-Man'i işaretli yoldan hayalete götürmek için aşağıdaki talimatlarda hangi adım eksiktir?



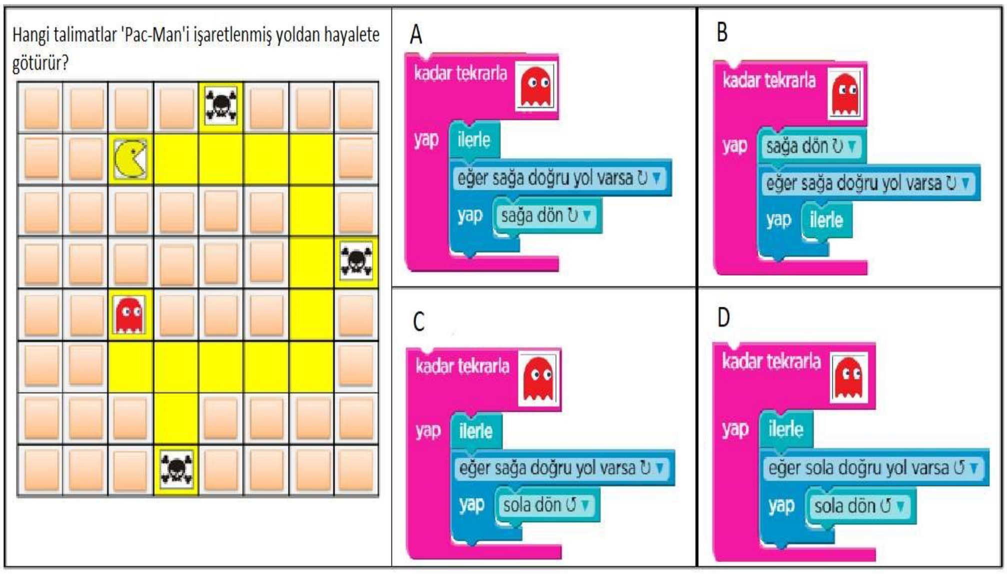
10. Soru: Talimatlar, 'Pac-Man'i işaretlenmiş yoldan hayalete götürmelidir. Talimatların hangi adımında bir hata vardır?



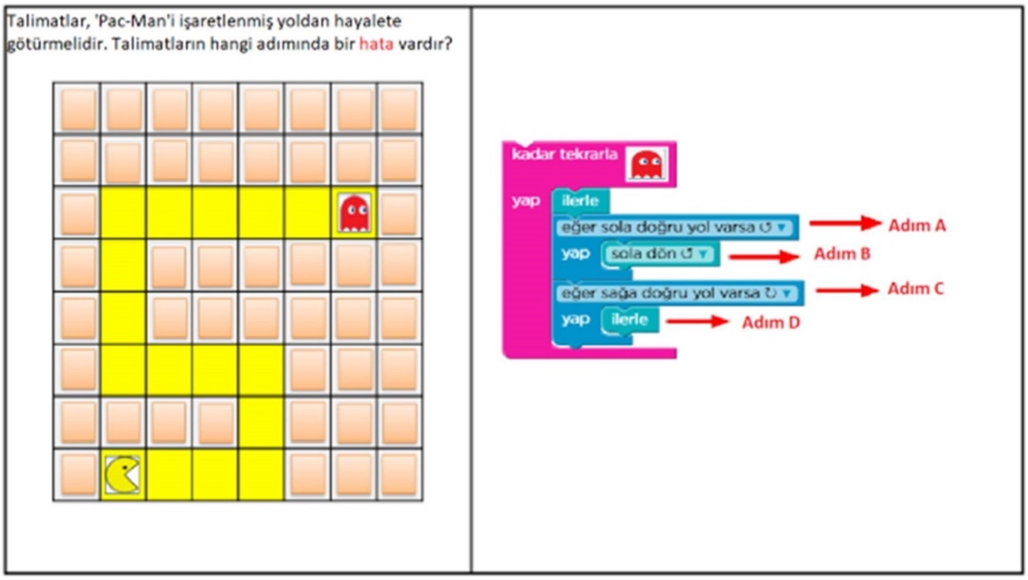
11. Soru: Hangi talimatlar Pac-Man'I işaretlenmiş yoldan hayalete götürür?



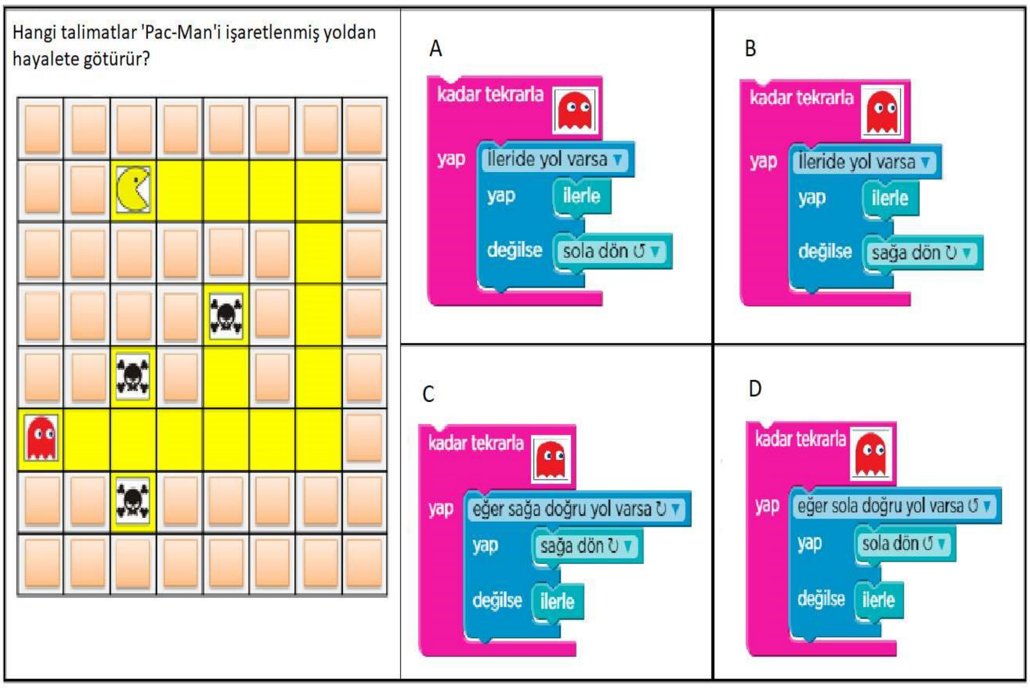
12. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldan hayalete götürür?



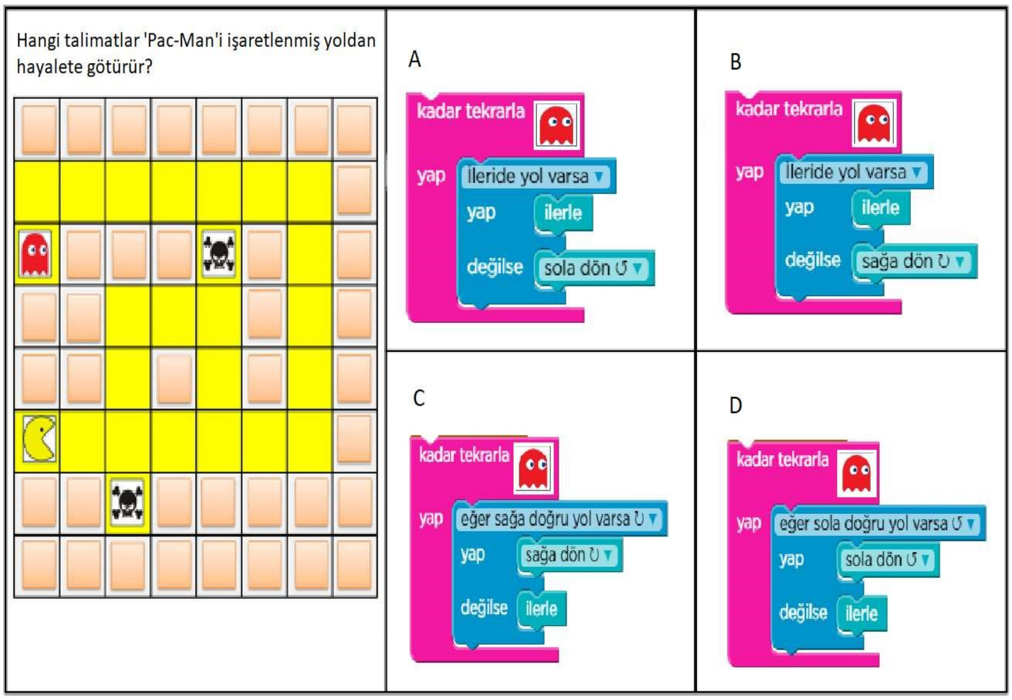
13. Soru: Talimatlar, 'Pac-Man'i işaretlenmiş yoldan hayalete götürmelidir. Talimatların hangi adımında bir hata vardır?



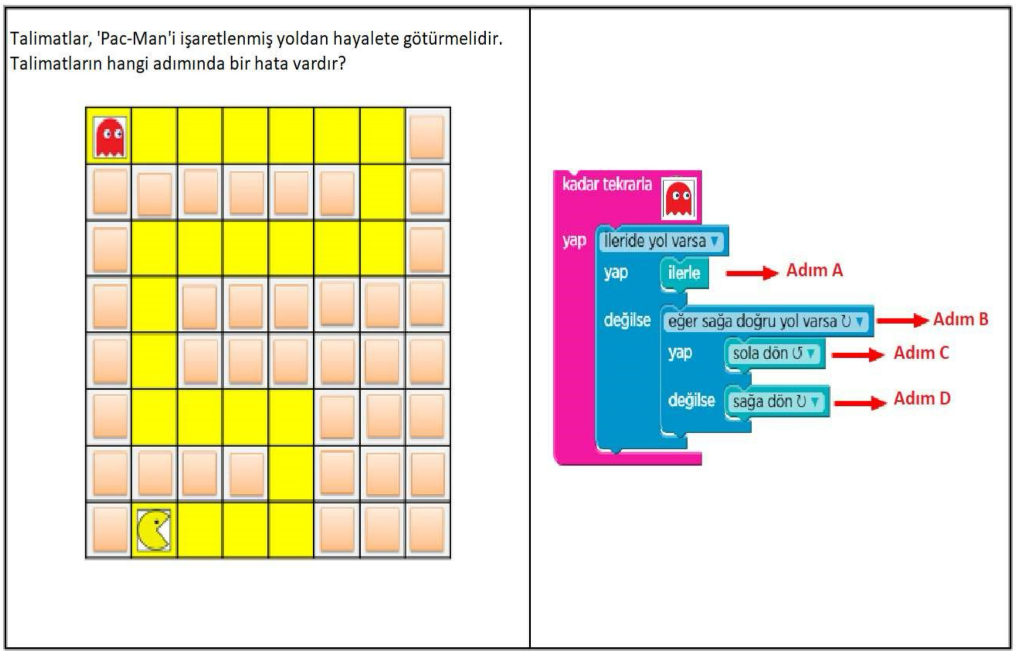
14. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldan hayalete götürür?



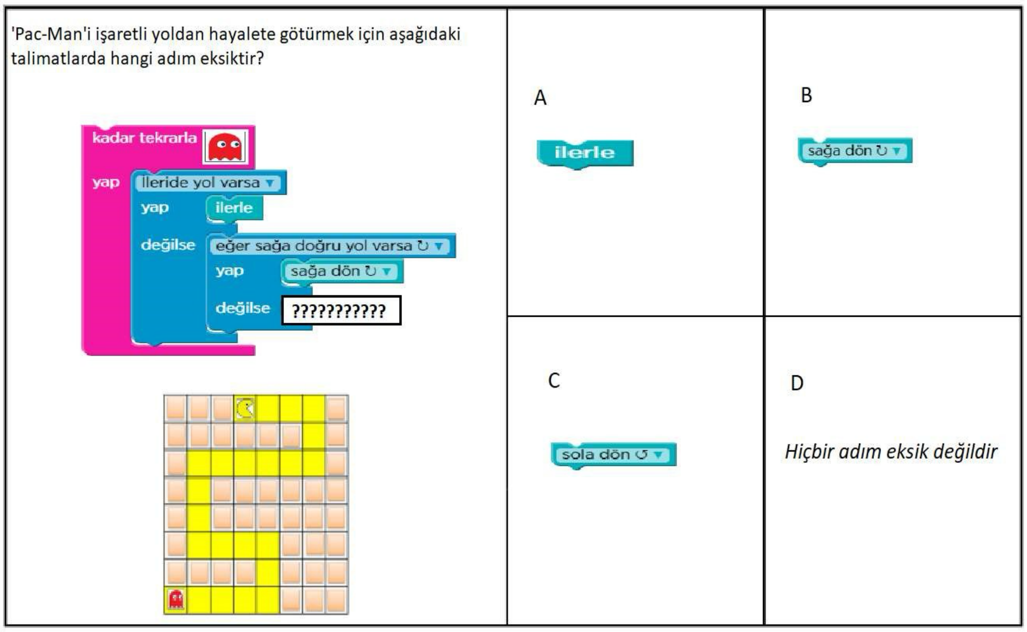
15. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldan hayalete götürür?



16. Soru: Talimatlar, 'Pac-Man'i işaretlenmiş yoldan hayalete götürmelidir. Talimatların hangi adımında bir hata vardır?

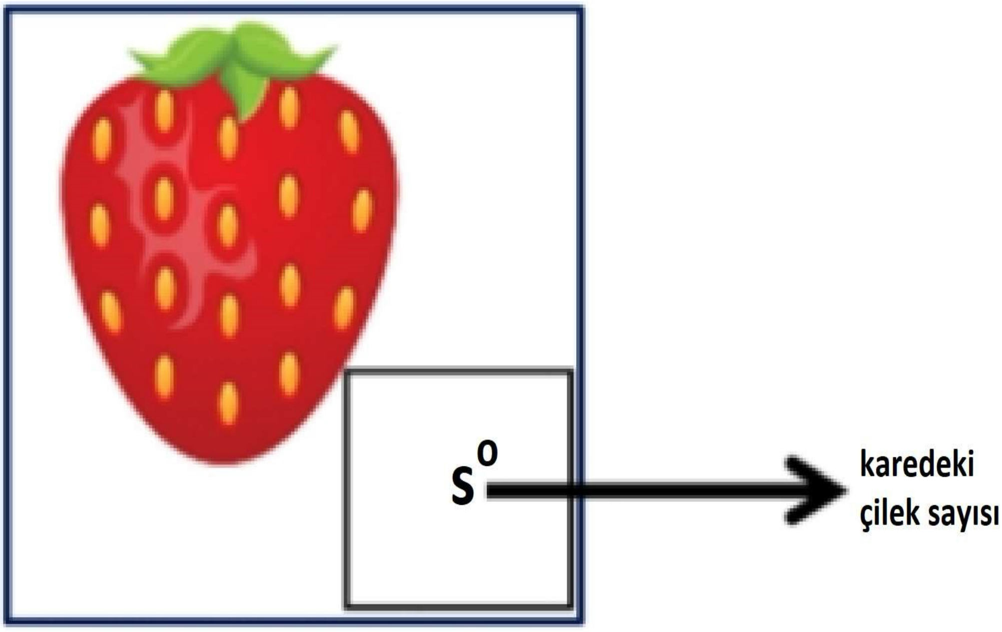


17. Soru: 'Pac-Man'i işaretli yoldan hayalete götürmek için aşağıdaki talimatlarda hangi adım eksiktir?

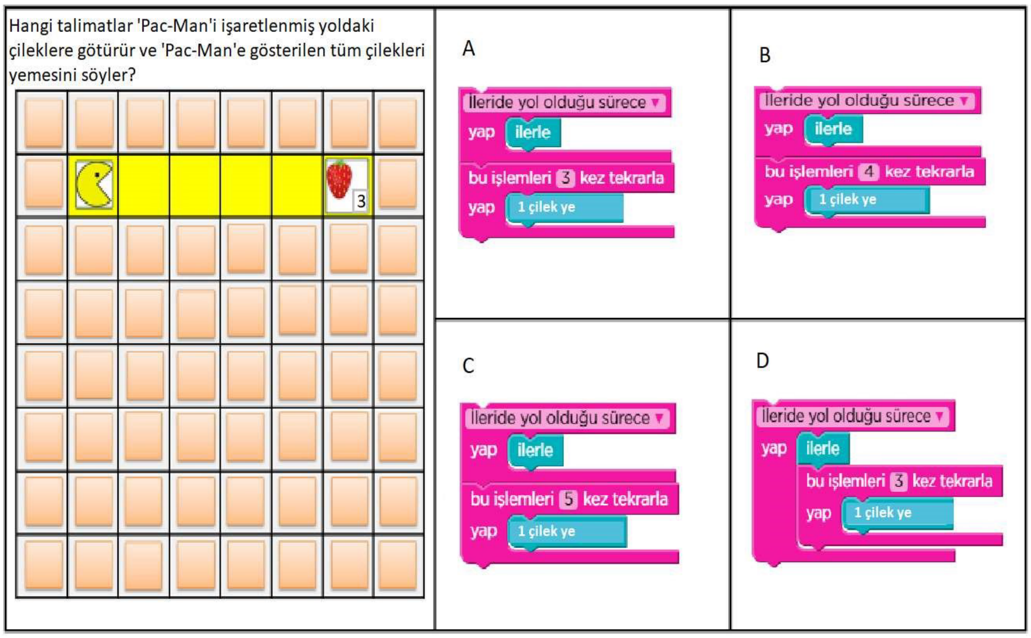


ÖNEMLİ : DİKKATLİCE OKUYUN

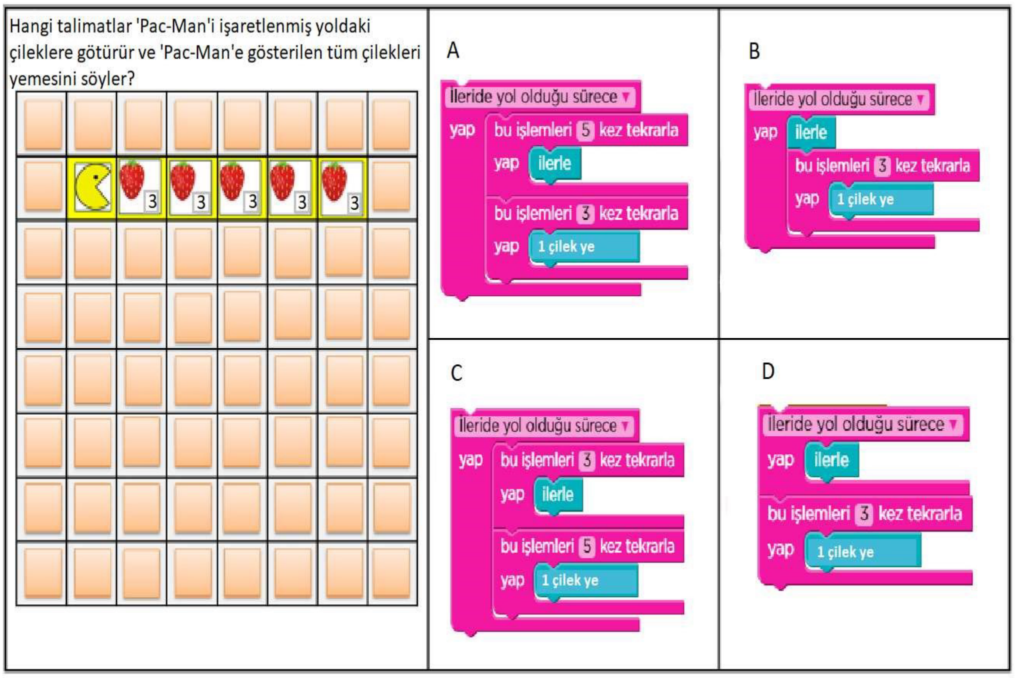
Bu soru grubunda, bazı karelerde çilek görüntüsü belirir. Resmin sağ alt köşesindeki sayı, karede kaç tane çilek olduğunu gösterir.



18. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldaki çileklere götürür ve 'Pac-Man'e gösterilen tüm çilekleri yemesini söyler?

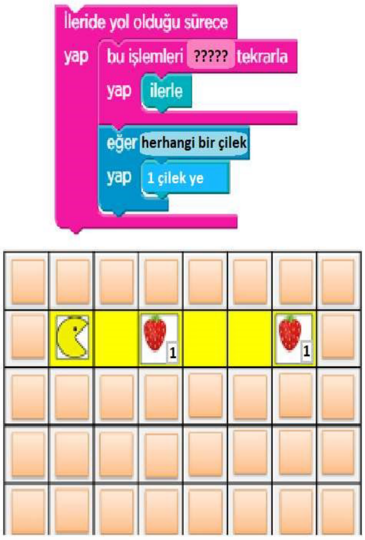


19. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldaki çileklere götürür ve 'Pac-Man'e gösterilen tüm çilekleri yemesini söyler?



20. Soru: 'Pac-Man'i işaretlenmiş yoldan çileklere götürmek ve 'Pac-Man'e gösterilen tüm çilekleri yemesini söylemek için aşağıdaki talimatlarda eksik olan nedir?

'Pac-Man'i işaretlenmiş yoldaki çileklere götürmek ve 'Pac-Man'e gösterilen tüm çilekleri yemesini söylemek için aşağıdaki talimatlarda eksik olan nedir?



A

1 kez

B

2 kez

C

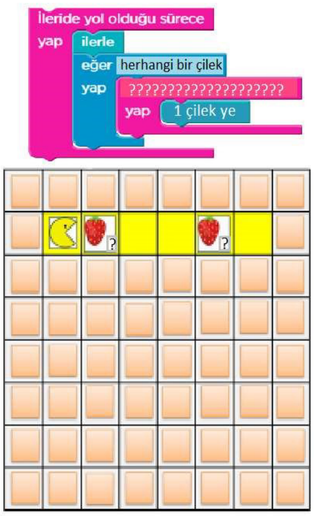
3 kez

D

5 kez

21. Soru: 'Pac-Man'i işaretlenmiş yoldan çileklere götürmek ve 'Pac-Man'e tüm çilekleri (bilinmeyen numara) yemesini söylemek için aşağıdaki talimatlarda hangi adım eksiktir?

'Pac-Man'i işaretlenmiş yoldan çileklere götürmek ve 'Pac-Man'e tüm çilekleri (bilinmeyen numara) yemesini söylemek için aşağıdaki talimatlarda hangi adım eksiktir?



A

ileride yol varken

B

ileride yol yokken

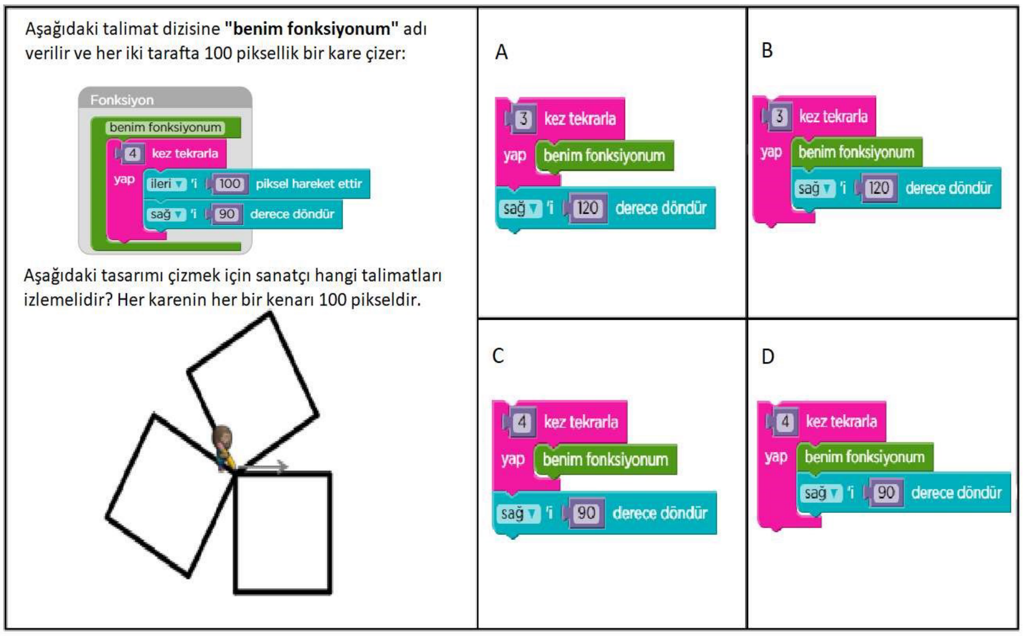
C

herhangi bir çilek varken

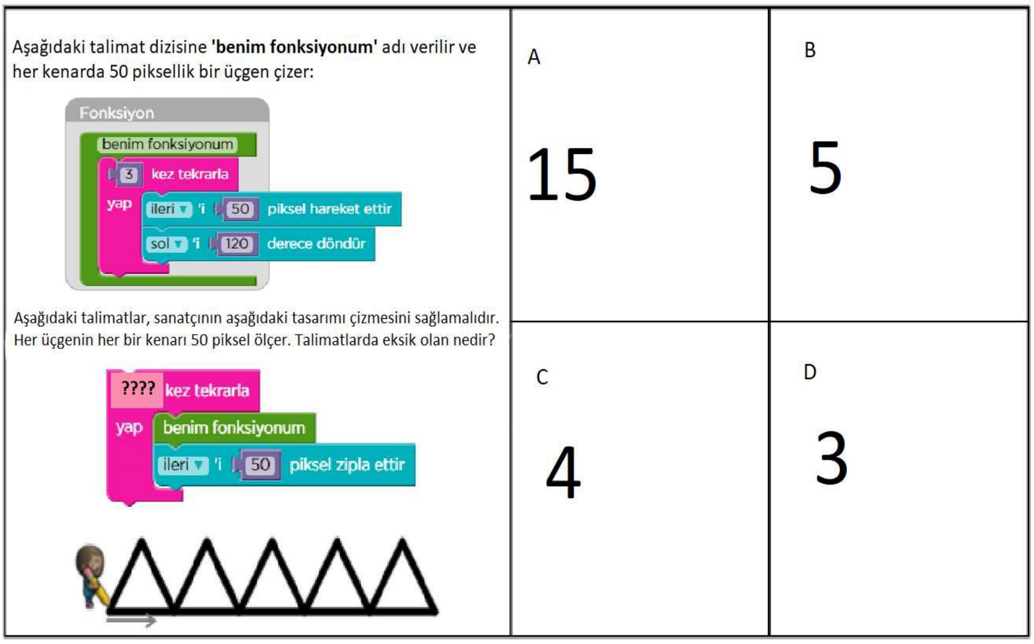
D

herhangi bir çilek yokken

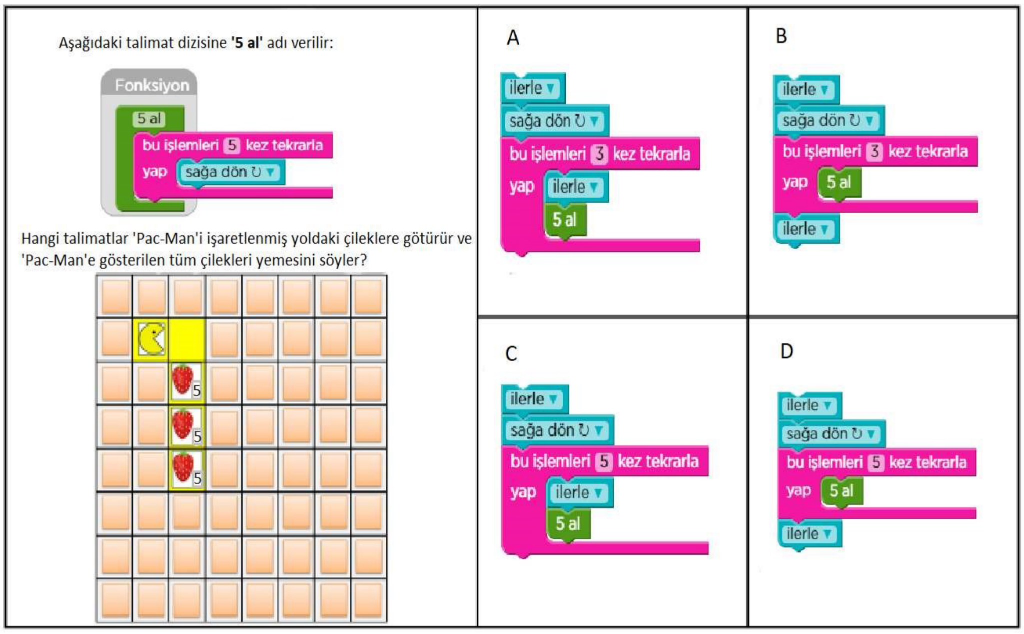
22. Soru: Aşağıdaki tasarımı çizmek için sanatçı hangi talimatları izlemelidir? Her karenin her bir kenarı 100 pikseldir.



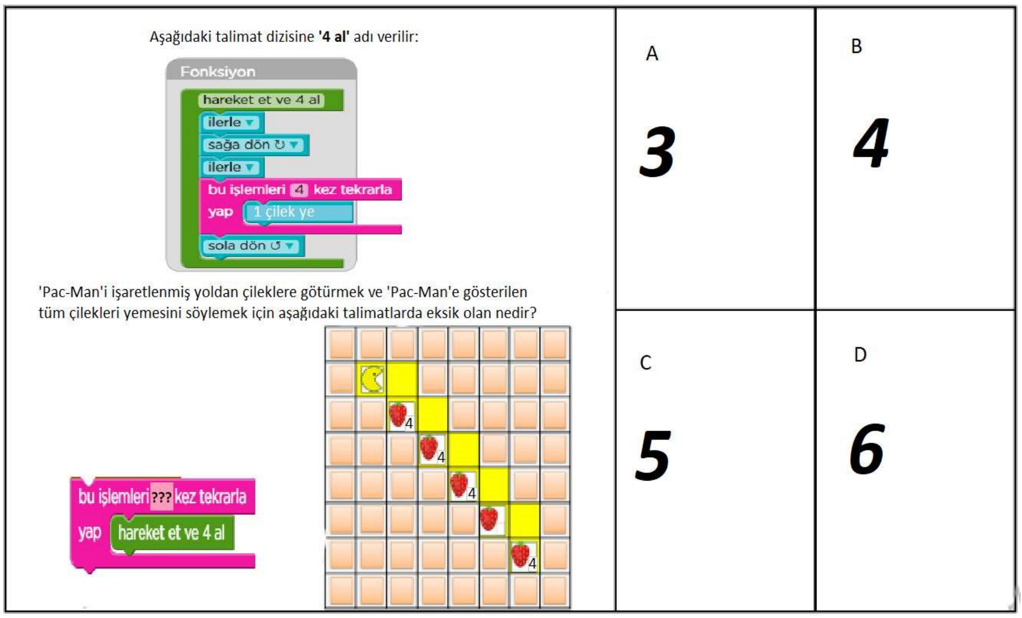
23. Soru: Aşağıdaki talimatlar, sanatçının aşağıdaki tasarımı çizmesini sağlamalıdır. Her üçgenin her bir kenarı 50 piksel ölçer. Talimatlarda eksik olan nedir?



24. Soru: Hangi talimatlar 'Pac-Man'i işaretlenmiş yoldaki çileklere götürür ve 'Pac-Man'e gösterilen tüm çilekleri yemesini söyler?



25. Soru: 'Pac-Man'i işaretlenmiş yoldan çileklere götürmek ve 'Pac-Man'e gösterilen tüm çilekleri yemesini söylemek için aşağıdaki talimatlarda eksik olan nedir?



C
ACTIVITY EVALUATION FORM

Etkinlik Görüş Formu

1. Scratch nedir?

2. Scratch ile neler yapıyorsunuz?

3. Scratch'i hayatınızın başka hangi alanlarında kullanabilirsiniz?

4. Scratch’te yaptığınız projelerinizde projenin fikrini nasıl buldunuz? Projenizi yapmaya nasıl başladınız?

5. Scratch’te proje oluşturma sürecinde zorlandınız mı? Zorlandığınızda ne yaptınız?

6. Scratch ile ilgili sevdiğiniz ve sevmediğiniz özellikler nelerdir? Scratch’te nelerin aynı kalmasını isterdiniz ve nelerin değişmesini isterdiniz?

7. Teknolojiyle ilgili yapmaktan hoşlandığınız diğer şeyler nelerdir?

D

THE PERMISSIONS OBTAINED FROM THE MINISTRY OF NATIONAL EDUCATION

T.C.
İSTANBUL VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : E-59090411-20-66010751 16/12/2022
Konu : Anket ve Araştırma İzni (Şükran SUNGUR)

VALİLİK MAKAMINA

İlgi : a) Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 21.01.2020 tarihli ve 2020/2 sayılı genelgesi.
b) Yıldız Teknik Üniversitesinin 02.11.2022 tarihli ve 2211020392 sayılı yazısı.
c) Müdürlüğümüz Araştırma ve Anket Komisyonunun 02.12.2022 tarihli tutanağı.

Araştırma Konusu	: The Effect of The Force and Motion Unit Integrated with Scratch Programming Activities on The Computatinal thinking Skills and Their Achievement of 6th Grade Middle School Students (Scratch Programlama Etkinlikleriyle Bütünleştirilen Kuvvet ve Hareket Ünitesinin Ortaokul 6. Sınıf Öğrencilerinin Bilgi - İşlemsel Düşünme Becerileri ve Başarılarına Etkisi)
Araştırma Türü	: Anket
Araştırma Yeri	: Esenyurt Recep Tayyip Erdoğan İmam Hatip Ortaokulu
Araştırma Kişiler	: Ortaokul Öğrencileri
Araştırmanın Süresi	: 2022 - 2023 Eğitim - Öğretim Yılı

Yukarıda bilgileri verilen araştırmanın; 6698 sayılı Kişisel Verilerin Korunması Kanununa aykırı veri istenmemesi, öğrenci velilerinden açık rıza onayı alınması, Covid-19 tedbirlerinin araştırmacı ve ilgili kurum idarelerince alınması, bilimsel amaç dışında kullanılmaması, bir örneği Müdürlüğümüzde muhafaza edilen mühürlü ve imzalı veri toplama araçlarının kurumlarımıza araştırmacı tarafından ulaştırılarak uygulanması, katılımcıların gönüllülük esasına göre seçilmesi, araştırma sonuç raporunun kamuoyuyla paylaşılmaması ve araştırma bittikten sonra 2 (iki) hafta içerisinde Müdürlüğümüze gönderilmesi, okul idarelerinin denetim, gözetim ve sorumluluğunda, eğitim ve öğretimi aksatmayacak şekilde, ilgi (a) genelge esasları dâhilinde uygulanması kaydıyla Müdürlüğümüzce uygun görülmektedir.

Makamınızca da uygun görüldüğü takdirde olurlarınıza arz ederim.

Levent YAZICI
İl Millî Eğitim Müdürü

OLUR
Dr. Hasan Hüseyin CAN
Vali a.
Vali Yardımcısı

Ek:
1- İlgi (b) Yazı ve Ekleri (38 Sayfa)
2- İlgi (c) Tutanak (1 Sayfa)

Adres : Binbirdirek Mah. İmran Öktem Cad. No: 1 Sultanahmet Fatih İstanbul Belge Doğrulama : https://www.turkiye.gov.tr/meb-ebys
Telefon : 0212 384 36 30 Bilgi İçin : Aydın BALTA
E-posta : stratejigelistirme34@meb.gov.tr Unvanı : VHKİ
Kep Adresi : meb@hs01.kep.tr İnternet Adresi : http://istanbul.meb.gov. tr/

E
LESSON PLANS

1st LESSON PLAN

LES son: Science
Grade Level: 6
Topic: Kuvvet ve Hareket / Bileşke Kuvvet
Süre: 40 + 40 min
Objective: F.6.3.1.1. Bir cisme etki eden kuvvetin yönünü, doğrultusunu ve büyüklüğünü çizerek gösterir.
Student Objectives: Problem çözme sürecinde takip edilmesi gereken adımları fark eder. Günlük hayatta karşılaştığı problemlere çözüm önerileri getirir. Verilen bir problemi uygun adımları kullanarak çözer. Verilen bir problemi analiz eder.
Expected pre-knowledge about topic: F.4.3.1.1. Kuvvetin, cisimlere hareket kazandırmasına ve cisimlerin şekillerini değiştirmesine yönelik deneyler yapar. F.5.3.1.1. Kuvvetin büyüklüğünü dinamometre ile ölçer.
Öğretme Öğrenme Yöntem ve Teknikleri:
Science Practices: Bilgiyi keşfederek bulma ve değerlendirme, argümantasyon yapma, bilgiyi inşa etme.
Kullanılan Eğitim Teknolojileri-Araç Gereçler: Bilgisayar/tablet/cep telefonu, Google Forms, Powerpoint,

GİRİŞ: Öğretmen kuvvetin ne olduğunu ve cisimler üzerindeki etkisini öğrencilere hatırlatmak için "Kuvvet nedir ve cisimler üzerinde ne gibi değişiklikler yapar" diye sorar. Öğrencilerden kuvvetin cisimlerin şeklini değiştirdiğini, duran cisimleri harekete geçirdiğini, hareketli cisimlerin durmasını sağladığını, yönünü ve süratini değiştirdiğini ve cisimleri döndürdüğünü cevapları beklenir. Öğrenciler yanlış bir nokta söylerse öğretmen direkt doğru veya yanlış olduğunu söylemeden öğrenciye soru sorarak öğrenciyi düşünmeye sevk eder. Öğrencilerden beklenen cevapları vermezse, öğretmen kuvvetin etkilerini hatırlatır. Bu hatırlatmadan sonra öğretmen sınıfı 4 kişilik gruplara ayırır ve senaryolaştırarak hazırlamış olduğu problem durumunun çalışma kağıdını (**Kaynakça A)** öğrencilere dağıtır ve keşfetmeye geçer.
KEŞFETME: Öğretmen dağıtmış olduğu çalışma kağıdının ekran görüntüsünü ekranında açar ve öğrencilere gösterir. Öğretmen öğrencilere bu problem durumu üzerinde çalışmaları için 10 dakika süreleri olduğunu belirtir Gruplar çalışırken öğretmen de gruplar arsında dolaşacaktır. Çalışma kağıdının son sorusunda verilen "Emre ve Aylin'in karşılaştığı günlük yaşam problemi olan ağır bir yükü Scrath programı üzerinde gösterip bu cismi hareket ettirmeyi sağlayabilir miyiz?" sorusu öğrencileri Scratch programına hazırlayacaktır. Öğrenciler soruları yanıtladıktan sonra öğretmenin daha önce hazırlamış olduğu Scratch çalışmasını (scratch öğretmen çalışması) öğrencilerle paylaşır ve ekranda açar. Burada öğrencilere verilen talimatlara göre öğrencilerden bireysel olarak proje üzerinde çalışmalarını ister. Çalışmada "Ekranda farklı renklerle verilmiş oklarla hareketsiz duran cismi istediğiniz yönde hareket ettirebilirsiniz." talimatı yazılmıştır. Öğretmen daha sonra bu çalışma ile ilgili çalışma kağıdını **(KAYNAKÇA B)** öğrencilere dağıtır ve ekranında da açıp gösterir. Burada öğrencilerin grupça sorulara yanıt vermelerini ister. Öğretmen bu sırada grupları dolaşır ve öğrencilerin soruları varsa yardımcı olur. Çalışma bittikten sonra her gruptan bir öğrenciye söz hakkı vereceğini söyler.
AÇIKLAMA: Öğretmen ekranını paylaşır ve her gruptan bir kişiye söz hakkı verip cevaplarını yorumlamasını ister. Öğrencilerin açıklamalarında bir hata veya eksiklik görürse müdahale eder. Daha sonra öğretmen "Duran bir cismi harekete geçiren, hareket eden bir cismi durduran, cisimlerin şeklini değiştiren, hareketli

cisimlerde yön ve sürat değişikliğine sebep olan ve cisimleri döndürebilen etkiye kuvvet denir. Kuvvetin büyüklüğünün birimi Newton'dır. Newton, kısaca N ile gösterilir. Kuvvet yönlü bir büyüklüktür yani kuvvetin uygulama noktası, belirli bir büyüklüğü, yönü ve doğrultusu vardır. Kuvvetin büyüklüğü dinamometre ile ölçülür. Kuvvet F sembolü ile gösterilir" açıklamasını hazırladığı sunumdan (kuvvet ve özellikleri sunumu) anlatır.
DERİNLEŞTİRME: Açıklamadan sonra öğretmen burada öğrencilerden cismi Scratch programı üzerinde bir çalışma yaparak hareketsiz bir cismi hareket ettirmelerini ister. Bu cisme ait yön ve doğrultularını belirtmeleri ister. Öğrencilerin hazırladıkları çalışmayı öğretmen ve arkadaşları ile paylaşmalarını ister. Öğrenciler öğretmenle çalışmalarını paylaştıktan sonra öğretmen öğrencilerin hazırladıkları çalışmanın kod bloklarını kontrol eder ve öğrencileri neden bu kod bloğunu kullandınız diye sorar. Öğrencinin verdiği cevaba göre başka bir öğrenciye arkadaşın böyle düşünüyor sen bu konuda ne düşünüyorsun diye sorar. Böylelikle sınıfta tartışma ortamı sağlanır.
DEĞERLENDİRME: Öğretmen öğrencilere Nearpod (nearpod etkinlik) üzerinden hazırladığı etkinliği öğrencilere tanıtır. Burada öğretmen öncelikle öğrencilere Nearpodu tanıtır ve herkesin bireysel olarak çalışacağını söyler. Daha sonra burada öğrencilerden ne istediğini anlatır. Nearpod üzerinden öğretmenin hazırlamış olduğu etkinlikte Görsellerde bazı cisimlere etki eden kuvvetler gösterilmiştir. Öğrencilerden bu görsellerin karşılarına bu kuvvetlerin özelliklerini yazmalarını ister. Bazı görsellerde de özellikleri verilen kuvvetleri öğrencilerden görseller üzerinde çizerek göstermelerini ister. Öğretmenim buradan Nearpod uygulamasının kullanmasının sebebi hem öğrenciler teknolojik bir alet üzerinden çizim yapabilme şansı elde edebilecek hem de Öğretmen bütün öğrencilerin hem çizimlerini hem de vermiş oldukları cevabı etkileşimli tahtadan kontrol edebilme şansını yakalayacaktır. Öğrenciler soruları yanıtlarken öğretmen de bu sırada etkileşimli tahtadan öğrenci yanıtlarını kontrol eder. Öğretmen öğrencilerin cevaplarına bakarak neler öğrendiklerini ve herhangi bir yanlış anlamaları veya soruları olup olmadığını görmüş olur. Böylelikle bu çalışma öğretmeni bir sonraki ders için hazırlamış olur.

EK PLAN: Öğretmen öğrencilerle birlikte Scratch'te balon patlatma oyunu tasarlar ve öğrenciler kendi tasarladıkları oyunu oynar.

KAYNAKÇA A

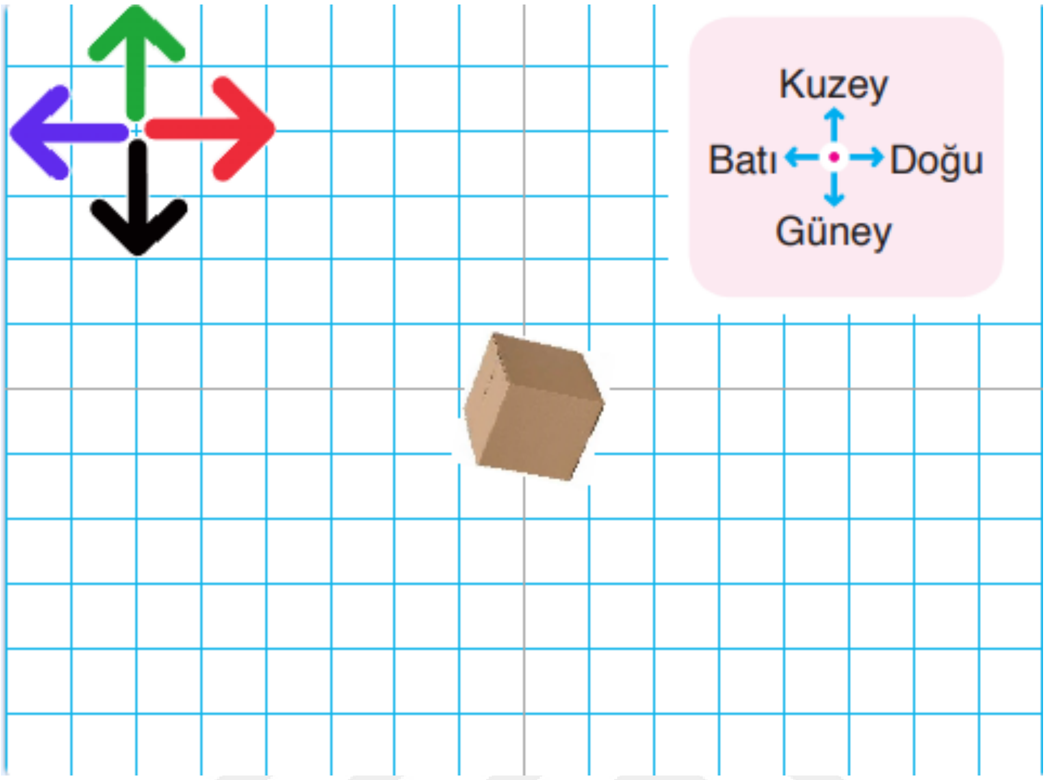
Çoğumuz bazen evimizdeki eşyaların düzeninden sıkılırız ve bu eşyaların yerlerini değiştirmek isteriz. Örneğin, televizyon, çalışma masamız ya da koltuklarımızın yerini değiştirdiğimizde evimizin havası da değişmiş olur ve bazen bu değişiklik bize iyi hissettirebilir. Sibel Hanım da evinin eski düzeninden sıkıldığı için evinde bir değişiklik yapmak ister ve koltukların yerini değiştirmeyi düşünür. Fakat koltuklar büyük ve ağır olduğu için Sibel Hanım tek başına yerden kaldırmakta güçlük çekmektedir. Bu durumda, Sibel Hanım'ın çocukları Emre ve Aylin koltuğun yerini değiştirmek için annelerine yardımcı olacaklardır. Fotoğrafta gördüğünüz *gibi Emre ve Aylin koltuğu ittikleri halde koltuğun hareket etmediğini gözlemliyorlar. Bu problem durumuna göre aşağıdaki soruları cevaplayınız.*

1. *Koltuğu yerinden kaldırmadan hareket ettirmek mümkün müdür? Mümkünse nasıl?*

2. *Koltuğun hareket etmemesindeki sebep ne olabilir? Emre ve Aylin'e bu problemi çözmelerine yardımcı olmak için nasıl bir çözüm önerirsiniz?*
3. *Koltuğun hareket etmesini sağlayan etkiye ne denir?*

4. *Emre ve Aylin'in karşılaştığı günlük yaşam problemi olan ağır bir yükü Scrath programı üzerinde gösterip bu cismi hareket ettirmeyi sağlayabilir miyiz?*

KAYNAKÇA B



1. *Uygulanan kuvvetlerin hepsi aynı büyüklükte midir? Cevabınızı açıklayınız.*

2. *Uygulanan kuvvetler farklı ise en büyük kuvvet hangi yönde uygulanmıştır? (Cevaplarınızı kuzey, güney, doğu, batı yönleri şeklinde belirtin.)*

3. *Hangi kuvvetler zıt yönlüdür?*

4. *Yönleri farklı olan kuvvetler aynı doğrultulu olabilir mi? Cevabınızı açıklayınız.*

2nd LESSON PLAN

Dersin Adı: Fen Bilimleri
Sınıf: 6
Ünite/Konu: Kuvvet ve Hareket / Bileşke Kuvvet
Süre: 40 + 40 +40 dk
Kazanım: F.6.3.1.2. Bir cisme etki eden birden fazla kuvveti deneyerek gözlemler.
Öğrenci Kazanımları:
Ön bilgiler: F.4.3.1.1. Kuvvetin, cisimlere hareket kazandırmasına ve cisimlerin şekillerini değiştirmesine yönelik deneyler yapar. F.6.3.1.1. Bir cisme etki eden kuvvetin yönünü, doğrultusunu ve büyüklüğünü çizerek gösterir.
Öğretme Öğrenme Yöntem ve Teknikleri:
Bilimsel Pratikler: Bilgiyi keşfederek bulma ve değerlendirme, argümantasyon yapma, bilgiyi inşa etme.
Kullanılan Eğitim Teknolojileri-Araç Gereçler: Bilgisayar/tablet/cep telefonu, Google Forms, Powerpoint,
GİRİŞ: Öğretmen, öğrencilerin derse olan ilgisini çekmek için ve konuya hazırlık için günlük hayatta öğrencilerin karşılaştığı bir oyun olan halat çekme oyununu hikayeleştirerek öğrencilere anlatır. Bu sırada öğretmenin ekranında da bu hikâye **(Kaynakça C)** açık olacaktır. Hikâyeyi anlattıktan sonra "İyide tüm bu halat çekme yarışında kazanan taraf ile kaybeden taraf arasındaki fark ne?" diye sorar. Beklenen cevap kazanan takımın daha fazla kuvvet uyguladığıdır ya

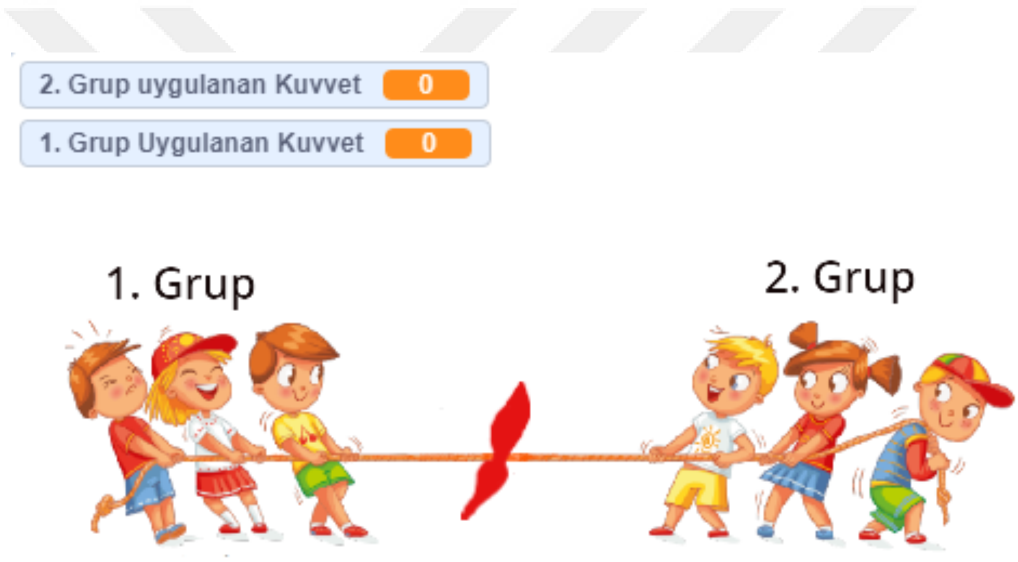
da kaybeden takımın daha az kuvvetle çektiğidir. Öğretmen öğrencilerden cevaplar toplar. Öğrenciler yanlış bir nokta söylerse öğretmen direkt doğru veya yanlış olduğunu söylemeden öğrenciye soru sorarak öğrenciyi düşünmeye sevk eder. Daha sonra öğretmen "Peki halat çekme oyununu Scratch uygulamasında nasıl tasarlayabilir der ve keşfetmeye geçer.
KEŞFETME: Öğretmen Scratch uygulamasından hazırlamış olduğu çalışmayı (scratch öğretmen çalışması)öğrencilerle paylaşır. Öğrencilerin bireysel olarak Scratch uygulamasında verilen talimatlara göre paylaşılan proje üzerinde çalışmalarını ister. Çalışmada "Halat çekme oyunu oynayan çocukların uyguladıkları kuvvetleri kendiniz belirleyerek oyunu kazanan takımı önce tahmin sonra tahminlerinizi test edin" talimatı yazılmıştır. Daha sonra öğretmen sınıfı 4 kişilik gruplara ayırır hazırlamış olduğu çalışma kağıdını (**Kaynakça D)** öğrencilere dağıtır. Dağıtmış olduğu çalışma kağıdının ekran görüntüsünü ekranında açar ve öğrencilere gösterir. Bu soruları cevaplamaları için 10 dakika süreleri olduğunu belirtir. Gruplar çalışırken öğretmen de gruplar arsında dolaşacaktır. Burada öğrencilerin grupça sorulara yanıt vermelerini ister. Öğretmen bu sırada grupları dolaşır ve öğrencilerin soruları varsa yardımcı olur. Çalışma bittikten sonra her gruptan bir öğrenciye söz hakkı vereceğini söyler.
AÇIKLAMA: Öğretmen ekranını paylaşır ve her gruptan bir kişiye söz hakkı verip cevaplarını sınıfla paylaşmalarını ister. Öğrencilerin açıklamalarında bir hata veya eksiklik görürse müdahale eder. Daha sonra öğretmen bileşke kuvvet ile ilgili hazırladığı sunumdan (bileşke kuvvet sunum) bir cisme etki eden farklı kuvvetleri ve bileşke kuvveti anlatır. Daha sonra konu ile ilgili Eba'dan video izletilir ve konu pekiştirilir.
DERİNLEŞTİRME: Açıklamadan sonra öğretmen burada öğrencilerden cismi Scratch programı üzerinde bir çalışma yaparak hareketsiz bir cisme birden fazla kuvvet uygulayarak hareket ettirmelerini ister. Öğretmenin daha önce paylaşmış olduğu çalışmadan farklı olarak değişkenlere bileşke kuvveti eklemelerini ve bileşke kuvveti girilen kuvvet değerlerine göre hesaplayıcı olarak belirlemelerini ister. Burada öğrencilerin soruları olursa öğretmen öğrencilere yardımcı olacaktır. Öğrencilerin hazırladıkları çalışmayı öğretmen ve arkadaşları ile

paylaşmalarını ister. Öğrenciler öğretmenle çalışmalarını paylaştıktan sonra öğretmen öğrencilerin hazırladıkları çalışmanın kod bloklarını kontrol eder ve öğrencileri neden bu kod bloğunu kullandınız diye sorar. Öğrencinin verdiği cevaba göre başka bir öğrenciye arkadaşın böyle düşünüyor sen bu konuda ne düşünüyorsun diye sorar. Böylelikle sınıfta tartışma ortamı sağlanır. Öğrencilerin Scratch'te çalışmaları bittikten sonra öğretmen hazırlamış olduğu çalışmayı (**Kaynakça E**) ekranda açar. Daha sonra öğrencilerden birinden verilenleri okumasını ister. Daha sonra başka bir öğrenciye sence verilenlerden hangisi kesinlikle doğrudur diye sorar. Öğrencinin verdiği cevaba göre başka bir öğrenciye arkadaşın böyle düşünüyor sen bu konuda ne düşünüyorsun diye sorar. Böylelikle sınıfta tartışma ortamı sağlanır.
DEĞERLENDİRME: Öğretmen öğrencilere google formlar üzerinden bir quiz (google formlar quiz) gönderir ve bu formu öğrencilerden doldurmalarını ister. Öğretmen de bu sırada ekranını paylaşır ve gelen yanıtları kontrol eder. Öğretmen öğrencilerin cevaplarına bakarak neler öğrendiklerini ve herhangi bir yanlış anlamaları veya soruları olup olmadığını görmüş olur. Böylelikle bu quiz öğretmeni bir sonraki ders için hazırlamış olur.
EK PLAN: Öğretmen öğrencilerle birlikte Scratch'te tenis oyunu tasarlar ve öğrenciler kendi tasarladıkları oyunu oynar.

KAYNAKÇA C

Antik Japon ve Çin imparatorluklarında toprak kazanmak için belirli bir zaman hiç savaş yapılmadı. Savaşmak yerine her iki taraf da en kuvvetli 250 savaşçısını seçti ve bu savaşçılar 167 metre uzunluğundaki halatın uçlarına geçip birbirlerini halat çekme oyununda yenmeye çalıştılar. Oyunu kazanan taraf hiç şiddet uygulamadan istediği toprağa elde edebiliyordu. İyide tüm bu halat çekme yarışında kazanan taraf ile kaybeden taraf arasındaki fark ne? Günümüzde bu gelenek her yıl 22 Eylül'de 3000 kişilik takımlarla tam 400 metrelik dev bir halatın uçlarında mücadele edilerek anılmaktadır. Savaşmadan, kaba kuvvet kullanmadan ve kazanmanın güzel bir yolu öyle değil mi?

KAYNAKÇA D

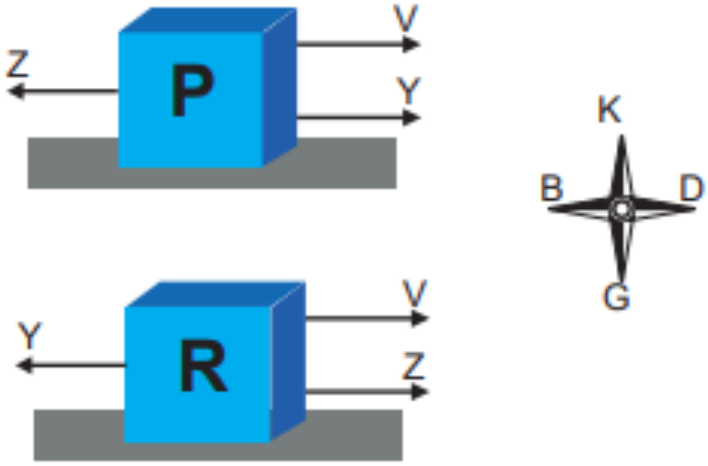


1. 1. grubun uyguladığı kuvvet 2. grubun uyguladığı kuvvetten daha büyük olduğunda halatta bulunan mendil hangi yöne doğru gitmiştir? (Yön ve doğrultularını belirtiniz) Neden? Cevaplarınızı açıklayınız.

2. 2. grubun uyguladığı kuvvet 1. grubun uyguladığı kuvvetten daha büyük olduğunda halatta bulunan mendil hangi yöne doğru gitmiştir? (Yön ve doğrultularını belirtiniz) Neden? Cevaplarınızı açıklayınız.

3. 1. grup ve 2. grubun uyguladığı kuvvetler birbirlerine eşit olduğunda mendilin hareketi hakkında ne söyleyebilirsiniz? Cevabınızı açıklayınız.

Aynı maddeden yapılmış, ağırlıkları aynı P ve R cisimlerine uygulanan kuvvetler aşağıda verilmiştir.



P cismi hareketsiz kalmaya devam ederken, R cismi doğu yönünde hareket etmeye başlamıştır.

Buna göre V, Y ve Z kuvvetlerinin büyüklükleri ile ilgili,

I. V, Y'den büyüktür.
II. Z, Y'den büyüktür.
III. V, Z'den küçüktür.

ifadelerinden hangileri <u>kesinlikle</u> doğrudur?

3rd LESSON PLAN

Dersin Adı: Fen Bilimleri
Sınıf: 6
Ünite/Konu: Kuvvet ve Hareket / Bileşke Kuvvet
Süre: 40 + 40 +40 dk
Kazanım: F.6.3.1.3. Dengelenmiş ve dengelenmemiş kuvvetleri, cisimlerin hareket durumlarını gözlemleyerek karşılaştırır.
Öğrenci Kazanımları:
Ön bilgiler: F.4.3.1.1. Kuvvetin, cisimlere hareket kazandırmasına ve cisimlerin şekillerini değiştirmesine yönelik deneyler yapar. F.6.3.1.2. Bir cisme etki eden birden fazla kuvveti deneyerek gözlemler. F.6.3.1.1. Bir cisme etki eden kuvvetin yönünü, doğrultusunu ve büyüklüğünü çizerek gösterir.
Öğretme Öğrenme Yöntem ve Teknikleri:
Bilimsel Pratikler: Bilgiyi keşfederek bulma ve değerlendirme, argümantasyon yapma, bilgiyi inşa etme.
Kullanılan Eğitim Teknolojileri-Araç Gereçler: Bilgisayar/tablet/cep telefonu, Google Forms, Powerpoint,
GİRİŞ: Öğretmen önceki derste işlenen bileşke kuvvetin sıfır olduğu halat çekme oyununu öğrencilere hatırlatmak ve dengelenmiş kuvvet konusuna geçiş yapmak için Scratch'te hazırladığı oyunu (**Kaynakça F)** öğrencilere göstererek "Bu halat çekme oyununda mendil ne sağa ne sola hareket etmektedir. Bunun sebebi nedir diye sorar." Öğrencilerden beklenen cevaplar bileşke kuvvetin sıfır olduğudur. Daha sonra öğretmen Scratch'te aynı oyunda farklı veriler girerek mendilin sağa veya sola hareket etmesini sağlar ve öğrencilere "Bu halat çekme

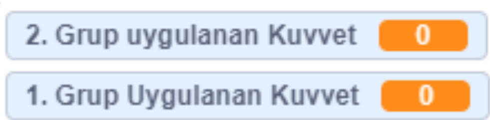
oyununda mendil girdiğimiz kuvvet değerlerine göre nasıl hareket etmiştir?" Öğrencilerden beklenen cevaplar mendil kuvvetin fazla olduğu yöne hareket etmiştir. Daha sonra öğretmen "Bunun sebebi nedir?" diye sorar. Burada öğrencilerden beklenen cevaplar bileşke kuvvet sıfırdan farklı olduğu içindir.
KEŞFETME: Daha sonra öğretmen sınıfı 4 kişilik gruplara ayırır ve çalışma kağıdını (**Kaynakça G)** öğrencilere dağıtır ve keşfetmeye geçer. Öğretmen dengelenmiş ve dengelenmemiş kuvvetlerin cisim üzerindeki etkilerini anlatmadan öğrencilerin şimdiye kadar öğrendiklerinden konu hakkında fikir yürütmelerini istediği için öğrencelerin çalışma kağıdını grupça yapmalarını ister. Dağıtmış olduğu çalışma kağıdının ekran görüntüsünü ekranında açar ve öğrencilere gösterir. Bu soruları cevaplamaları için 10 dakika süreleri olduğunu belirtir. Gruplar çalışırken öğretmen de gruplar arsında dolaşacaktır. Burada öğrencilerin grupça sorulara yanıt vermelerini ister. Öğretmen bu sırada grupları dolaşır ve öğrencilerin soruları varsa yardımcı olur. Çalışma bittikten sonra her gruptan bir öğrenciye söz hakkı vereceğini söyler.
AÇIKLAMA: Öğretmen ekranını paylaşır ve her gruptan bir kişiye söz hakkı verip cevaplarını sınıfla paylaşmalarını ister. Öğrencinin verdiği cevaba göre başka bir öğrenciye arkadaşın böyle düşünüyor sen bu konuda ne düşünüyorsun diye sorar. Böylelikle sınıfta tartışma ortamı sağlanır. Daha sonra öğretmen dengelenmiş ve dengelenmemiş kuvvetler konusunu hazırladığı sunumdan (dengelenmiş ve dengelenmemiş kuvvetler sunum) anlatır. Daha sonra öğrencilerden günlük hayatta dengelenmiş veya dengelenmemiş kuvvetler örnekleri vermeleri istenir. Beklenen cevaplar uçağın hareketi, paraşütün inmesi vb. Daha sonra konu ile ilgili video Eba'dan izletilerek konu pekiştirilir. Son olarak öğretmenin daha önce Scratch'ten hazırlamış olduğu oyunu (dengelenmiş ve dengelenmemiş kuvvetler) öğrencilerle paylaşır. Burada öğrencilerden kuvvet değerlerini değiştirerek cismin dengelenmiş ve dengelenmemiş olduğu durumları test etmelerini ister.
DERİNLEŞTİRME: Öğretmen günlük hayatta karşılaşılan bir durumu doğru ve yanlış yargılar içeren öğrenci yorumlarıyla hazırlamış olduğu çalışma kağıdını (**Kaynakça H**) öğrencilere dağıtır. Dağıtmış olduğu çalışma kağıdının ekran

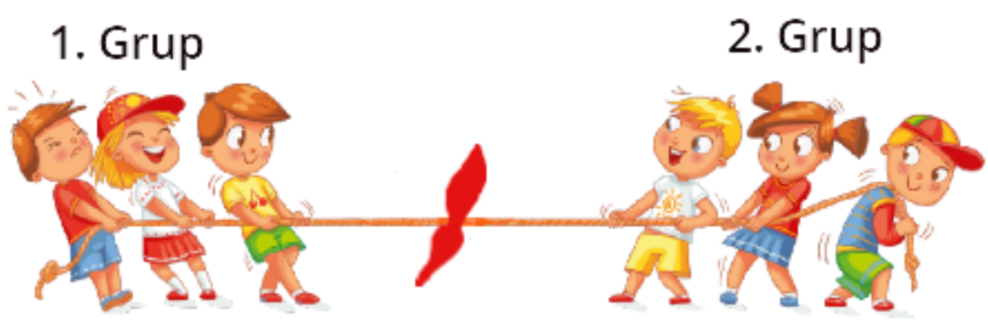
görüntüsünü ekranında açar ve öğrencilere gösterir. Burada öğrencilerin grupça sorulara yanıt vermelerini ister. Gruplar çalışırken öğretmen de gruplar arsında dolaşacaktır. Öğrenciler grupça çalışma kağıdını yaptıktan sonra her gruptan bir kişiye söz hakkı verilir. Grubun verdiği cevaba göre başka bir gruba bu grup böyle düşünüyor. Siz bu konu hakkında ne düşünüyorsunuz diye sorularak sınıfta tartışma ortamı sağlanır. Daha sonra öğretmen öğrencilerden Scratch üzerinden dengelenmiş ve dengelenmemiş kuvvetler ile ilgili bir oyun tasarlamalarını ister. Tüm gruplar çalışmaları bittikten sonra öğrenciler bu çalışmaları öğretmen ile paylaşır ve öğretmen ekranda öğrencilerin çalışmalarını kontrol eder.

DEĞERLENDİRME: Öğretmen öğrencilerin bu derste neler öğrendikleri ile alakalı bir değerlendirme yapmak için çıkış kartı olarak Wordwall üzerinden hazırladığı 5 soruluk bir quiz/ değerlendirme çarkı linkini Wardwall çark paylaşır. Öğrenciler öğretmen tarafından gruplara ayrılırlar ve çark oyununu oynar. Çarkları çevirir ve gelen soruları yanıtlayıp tartışırlar.

EK PLAN: Öğretmen öğrencilerle birlikte Scratch'te meyve yeme oyunu tasarlar ve öğrenciler kendi tasarladıkları oyunu oynar.

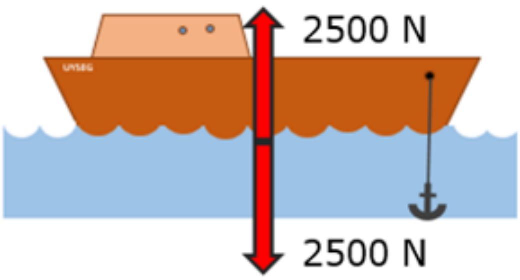
KAYNAKÇA F





KAYNAKÇA G

Kuvvetler

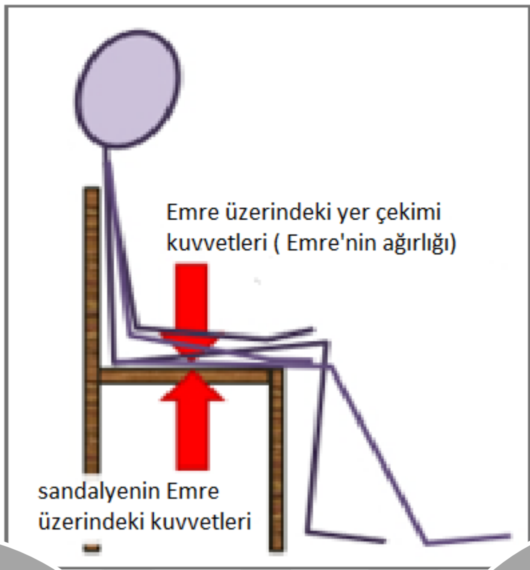


Sonra ne olur?

- Hızlanır
- Daha yüksek sabit hızda gider
- Aynı sabit hızda gider
- Durur
- Yavaşlar
- Daha yavaş sabit hızda gider
- Hareketsiz kalır
- Geriye gider

KAYNAKÇA H

Aşağıda bazı öğrencilerin Emre sandalyeye oturduğunda neler olduğuna dair bazı fikirleri verilmiştir.



Kaan

Kuvvetler hala ortada olduğu için koltuğun şeklini değiştirmişlerdir.

Zeynep

Emre hareket etmediği için kuvvet yoktur.

Ela

Sandalye, Emre'nin ağırlığından daha büyük bir kuvvetle yukarı itilir.

Ali

Emre oturduğunda Emre'nin üzerindeki kuvvetler dengelenir.

Sarp

Emre çantasını kucağına alırsa, kuvvetler daha büyük olacaktır.

1. Emre üzerindeki kuvvetler konusunda kim doğru söylemiştir?

2. Kim kuvvet konusunu tam olarak anlamamıştır? Anlamalarına yardımcı olmak için onlara ne söylerdiniz?

4th LESSON PLAN

Dersin Adı: Fen Bilimleri
Sınıf: 6
Ünite/Konu: Kuvvet ve Hareket / Sabit Süratli Hareket
Süre: 40 + 40 +40 dk
Kazanım: F.6.3.2.1. Sürati tanımlar ve birimini ifade eder.
Öğrenci Kazanımları:
Ön bilgiler:
Öğretme Öğrenme Yöntem ve Teknikleri:
Bilimsel Pratikler: Bilgiyi keşfederek bulma ve değerlendirme, argümantasyon yapma, bilgiyi inşa etme.
Kullanılan Eğitim Teknolojileri-Araç Gereçler: Bilgisayar/tablet/cep telefonu, Google Forms, Powerpoint,
GİRİŞ: Öğretmen öğrencilerin derse olan ilgisini çekmek ve öğrencileri konuya hazırlamak için "Dünya'nın en hızlı hayvanı sizce hangisidir?" diye sorar. Öğrencilerin cevaplarını topladıktan sonra derse TUBİTAK bilim çocuktan getirdiği haberi TUBİTAK Haber ekranında paylaşarak okur. Daha sonra "Günümüzde elimizdeki verilere göre karanın en hızlı hayvanı Maty'dır. Fakat bundan önce en hızlı hayvan Avustralya Kaplan böceğiydi. Bu durum bize neyi gösterir?" Öğretmen burada öğrencilerin bilimin doğası üzerinde düşündürmeyi hedefler. Öğrencilerden beklenen cevaplar bilimin sabit

kalmadığı bilimin değiştiği ve geliştiğidir. Daha sonra “Peki sizce hız nedir?” diye sorar. Öğrencilerden cevaplar alındıktan sonra herhangi bir açıklama yapmadan “Hadi bunu bir çocukken çoğumuzun oynadığı saklambaç oyunu etkinliğiyle anlamaya çalışalım” der ve keşfetmeye geçer.
KEŞFETME: Daha sonra öğretmen sınıfı 4 kişilik gruplara ayırır ve çalışma kağıdını (**Kaynakça I)** öğrencilere dağıtır. Burada öğretmen çoğu öğrencinin çocukken oynamış olduğu saklambaç oyunu ile ilgili bir çalışma hazırlamıştır. Öğrenciler bu oyunu düşünerek sorulara cevap vermeye çalışacaklardır. Dağıtmış olduğu çalışma kağıdının ekran görüntüsünü ekranında açar ve öğrencilere gösterir. Bu soruları cevaplamaları için 15 dakika süreleri olduğunu belirtir. Gruplar çalışırken öğretmen de gruplar arsında dolaşacaktır. Burada öğrencilerin grupça sorulara yanıt vermelerini ister. Öğretmen bu sırada grupları dolaşır ve öğrencilerin soruları varsa yardımcı olur. Çalışma bittikten sonra her gruptan bir öğrenciye söz hakkı vereceğini söyler.
AÇIKLAMA: Öğretmen ekranını paylaşır ve her gruptan bir kişiye söz hakkı verip cevaplarını sınıfla paylaşmalarını ister. Öğrencinin verdiği cevaba göre başka bir öğrenciye arkadaşın böyle düşünüyor sen bu konuda ne düşünüyorsun diye sorar. Öğrenciler cevaplarını paylaştıktan öğretmen süratin tanımını yapar ve birimlerini söyler. Daha sonra konunun pekişmesi için Eba’dan konu ile ilgili video izletilir. Daha sonra sunumdan sürat sunum hazırladığı etkinlikle sürati pekiştirmek için öğrencilere sorular sorarak öğrencileri yönlendirir ve öğretmen gerekli açıklamaları yapar.
DERİNLEŞTİRME: Öğretmen hem konuyu derinleştirmek hem de Fen eğitiminde bağımlı değişken, bağımsız değişken ve kontrol edilen değişken konusunu öğrencilere kavratmak amacıyla hazırladığı etkinliği (**Kaynakça K**) öğrencilerle paylaşır. Öğretmen sınıfı 4 kişilik gruplara ayırır ve çalışma kağıdını öğrencilere dağıtır. Dağıtmış olduğu çalışma kağıdının ekran görüntüsünü ekranında açar ve öğrencilere gösterir. Bu soruları cevaplamaları için 10 dakika süreleri olduğunu belirtir. Gruplar çalışırken öğretmen de gruplar arsında dolaşacaktır. Burada öğrencilerin grupça sorulara yanıt vermelerini ister. Öğretmen bu sırada grupları dolaşır ve öğrencilerin soruları varsa

yardımcı olur. Çalışma bittikten sonra her gruptan bir öğrenciye söz hakkı vereceğini söyler. Öğretmen ekranını paylaşır ve her gruptan bir kişiye söz hakkı verip cevaplarını sınıfla paylaşmalarını ister. Öğrencinin verdiği cevaba göre başka bir öğrenciye arkadaşın böyle düşünüyor sen bu konuda ne düşünüyorsun diye sorar. Öğrenciler cevaplarını paylaştıktan sonra öğretmen bu kavramların tanımlarını yapar ve başka bir örnek sorarak öğrencilerden buradaki değişkenleri belirlemelerini ister. Daha sonra konunun pekişmesi için öğretmen öğrencilerle Scratch üzerinden tasarladığı çalışmayı Scratch proje paylaşır. Öğrencilere talimatlara göre verileri girip neler gözlemlediklerini sorar. Girilen verilere göre arabanın hareketinin nasıl değiştiği sorulur. Daha sonra öğrencilerden sürat ile öğrendikleri kavramlara göre Scratch üzerinden bir proje tasarlamalarını ve sınıfla paylaşmalarını ister.
DEĞERLENDİRME: Öğretmen öğrencilere Nearpod üzerinden hazırladığı etkinliği öğrencilere tanıtır. Daha sonra burada öğrencilerden ne istediğini anlatır. Nearpod üzerinden öğretmenin hazırlamış olduğu etkinlikte öğrenciler soruya Poll üzerinden cevap vereceklerdir. Öğretmenim buradan Nearpod uygulamasının kullanmasının sebebi öğretmen bütün öğrencilerin vermiş oldukları cevabı etkileşimli tahtadan kontrol edebilme şansını yakalayacaktır. Üstelik Poll sınıfın kaçta kaçının hangi şıkkı işaretlediğini gösterdiği için öğrencileri yanıltan veya güçlü çeldiricileri öğretmen ders esnasında görebilme şansı yakalayacaktır. Öğrenciler soruları yanıtlarken öğretmen de bu sırada etkileşimli tahtadan öğrenci yanıtlarını kontrol eder. Öğretmen öğrencilerin cevaplarına bakarak neler öğrendiklerini ve herhangi bir yanlış anlamaları veya soruları olup olmadığını görmüş olur. Böylelikle bu çalışma öğretmeni bir sonraki ders için hazırlamış olur.
EK PLAN: Öğretmen öğrencilerle birlikte Scratch'te hesap makinası tasarlar ve öğrenciler kendi tasarladıkları hesap makinesiyle oynar.

KAYNAKÇA I

KAYNAKÇA K

Minel bir bilgisayar programı kullanarak sanal bir deney benzetimi yapıyor. Bu benzetimde bir araba çizen Minel çizdiği bu araba ile deneye başlamadan önce şunları belirlemesi gerekiyor.

1. Arabanın harekete başladığı sürat
2. Arabanın hareketine zıt yönde uygulanacak sürtünme kuvveti
3. Arabanın hareket edeceği süre

Benzetim sona erdiğinde program arabanın gittiği yolu hesaplıyor. Minel'in hazırladığı bazı deney benzetimleri gösterilmiştir.

Minel sabit süratli bir cismin sürati ile alınan yol arasındaki ilişkiyi göstermek için hangi iki benzetimin sonuçlarını karşılaştırmalıdır?

5th LESSON PLAN

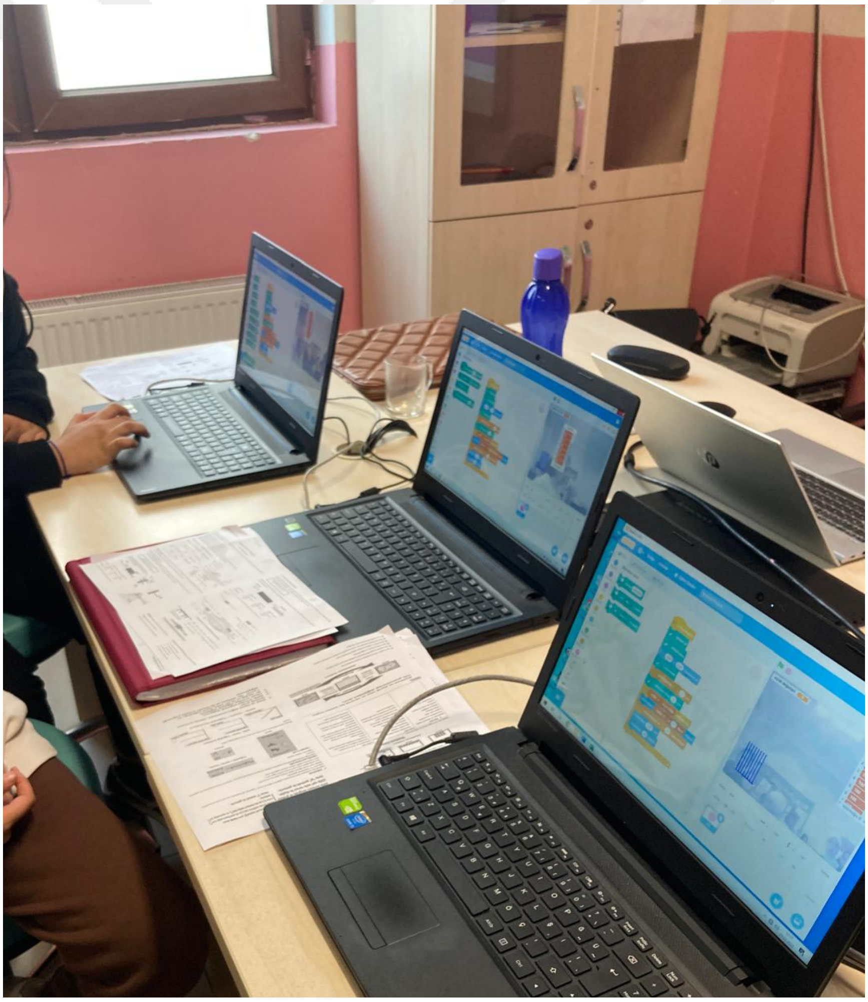
Dersin Adı: Fen Bilimleri
Sınıf: 6
Ünite/Konu: Kuvvet ve Hareket / Sabit Süratli Hareket
Süre: 40 + 40 +40 dk
Kazanım: F.6.3.2.2. Yol, zaman ve sürat arasındaki ilişkiyi grafik üzerinde gösterir.
Öğrenci Kazanımları:
Ön bilgiler:
Öğretme Öğrenme Yöntem ve Teknikleri:
Bilimsel Pratikler: Bilgiyi keşfederek bulma ve değerlendirme, argümantasyon yapma, bilgiyi inşa etme.
Kullanılan Eğitim Teknolojileri-Araç Gereçler: Bilgisayar/tablet/cep telefonu, Google Forms, Powerpoint,
GİRİŞ: Öğretmen öğrencilerin derse olan ilgisini çekmek ve konuyla günlük yaşamı ilişkilendirmek için yakın zamanda dünyanın dönme hızıyla ilgili haberi (haber) paylaşır. Habere göre 29 Haziran'da Dünya'nın şimdiye kadar kaydedilen en kısa günüdür. Öğrencilere bunun sebebinin ne olabileceği ve bu dönüş hızındaki değişimin etkisin neler olabileceği sorulur. Öğrencilerden cevaplar alındıktan sonra haberin devamında bilim adamlarının açıklamaları okunur. Daha sonra öğrencileri konuya hazırlamak için "Dünyanın kendi ekseni etrafında saatte bin 700 kilometre hızla döndüğünü belirtiliyor ve bu ses hızından daha yüksektir. Ancak bizler yüzeyde yaşayan canlılar olarak bu hızı hissetmeyiz. Bunun nedeni ne olabilir?" diye sorulur. Öğrenci tahminleri alındıktan sonra bu durumun sebebinin Dünya'nın neredeyse sabit bir hızla

döndüğü açıklaması yapılır. Daha sonra öğretmen hazırladığı sunumdan (sürat sunum) sabit süratli bir cismin yol-zaman ve sürat-zaman grafiğinin nasıl çizildiğini anlatır. "Peki bir cismin aldığı yolu, sürati ve geçen zamanı bilgisayar ortamında grafik üzerinde gösterebilir miyiz?" diye sorar. Öğrenciler daha önce bilgisayar ortamında Nearpod üzerinden çizim yaptığı için öğrencilerden evet Nearpod üzerinden yapılabilir cevapları beklenir. Burada öğrencilerin bildiği başka bir uygulama da varsa öğretmen paylaşmalarını ister ve keşfetmeye geçer.
KEŞFETME: Öğretmen öğrencilere Nearpod üzerinden hazırladığı etkinliği (nearpod grafik) öğrencilere tanıtır ve herkesin bireysel olarak çalışacağını söyler. Daha sonra burada öğrencilerden ne istediğini anlatır. Nearpod üzerinden öğretmenin hazırlamış olduğu etkinlikte görselde bir cismin zamana bağlı aldığı yol gösterilmiştir. Öğrencilerden yol ve zamana bakarak cismin süre boyunca süratini hesaplayıp yazmalarını ister. Diğer iki soru ise cisme ait yol-zaman ve sürat-zaman grafiklerini çizmelerini ister. Öğrenciler soruları yanıtlarken öğretmen de bu sırada etkileşimli tahtadan öğrenci yanıtlarını kontrol eder. Öğretmen öğrencilerin cevaplarına bakarak neler öğrendiklerini ve herhangi bir yanlış anlamaları veya soruları olup olmadığını görmüş olur. Böylelikle bu çalışma öğretmeni bir sonraki ders için hazırlamış olur.
AÇIKLAMA: Öğretmen öğrencilerin çizimlerinde veya cevaplarında bir hata ya da eksiklik varsa açıklar. Daha sonra öğretmen de soruların cevaplarını söyler ve açıklar. Konunun pekişmesi için öğrencilerle Eba'da verilmiş olan simülasyon yapılır.
DERİNLEŞTİRME: Öğretmen öğrencilerin Scratch üzerinden grafik çizebilmeleri için yapmış olduğu çalışmayı (scratch grafik) ve kod bloklarını öğrencilere gösterir. Scratch üzerinden öğrenciye verilmiş olan çalışma kağıdındaki yol ve zamana bakarak öğrencilerin bireysel olarak bu cismin sürat grafiğini çizmelerini ister. Öğretmen bu esnada sınıfı dolaşır ve öğrencilerin soruları varsa yanıtlar. Daha sonra öğrencilerden çalışmalarını sınıfla paylaşmalarını ve çizimlerini açıklamalarını ister.

DEĞERLENDİRME: Öğretmen öğrencilere google formlar üzerinden bir quiz (google formlar) gönderir ve bu formu öğrencilerden doldurmalarını ister. Öğretmen de bu sırada ekranını paylaşır ve gelen yanıtları kontrol eder. Öğretmen öğrencilerin cevaplarına bakarak neler öğrendiklerini ve herhangi bir yanlış anlamaları veya soruları olup olmadığını görmüş olur. Böylelikle bu quiz öğrencilerin derste neler öğrendiğine dair öğretmene fikir verir.
EK PLAN: Öğretmen öğrencilerle birlikte Scratch'te Dünya ve Ay'ın dönme ve dolanma hareketlerini tasarlar.

F

IMAGES OF THE STUDENTS TAKEN DURING STUDY



PUBLICATIONS FROM THE THESIS

Conference Papers

1. Sungur, Ş., & Ozkan, G. (2024). Middle school students' opinions about Scratch programming activities. *BZT TURAN PUBLISHING HOUSE*, 29–31.