T.C.
MANISA CELAL BAYAR UNIVERSITY
GRADUATE SCHOOL OF EDUCATION

MASTER'S THESIS
DEPARTMENT OF COMPUTER ENGINEERING
COMPUTER ENGINEERING PROGRAM

AN EDUCATIONAL GAME DESIGN FOR ROBOTIC CODING

Esma ÇELİKTEN

Supervisor
Doç. Dr. Birim BALCI

MANISA – 2025

Esma
ÇELİKTEN

AN EDUCATIONAL GAME DESIGN FOR ROBOTIC CODING

2025

COMMITMENT

I hereby declare that this thesis has been written in accordance with the academic and ethical rules in Manisa Celal Bayar University Graduate School of Postgraduate Education, Department of Computer Engineering, and that all the literature information used in the thesis is included in the thesis with reference, that it is entirely my own work, that I have cited references for each citation, and that I have not used any artificial intelligence and programmes contrary to academic and ethical rules in the writing of the thesis.

Esma ÇELİKTEN

ÖZET

Yüksek Lisans Tezi

Esma Çelikten

Manisa Celal Bayar Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman:
Doç. Dr. Birim BALCI

Teknolojik ilerlemeler gelişmeye devam ettikçe, yeni öğrenme yöntemlerine erişim giderek daha çeşitli hale geldi. Öğrenme artık kitaplarla veya gazetelerle sınırlı kalmayıp çeşitli dijital platformlara yayılıyor. Bu platformlar öğrenmeyi daha erişilebilir, ilgi çekici ve akılda kalıcı hale getirmeyi amaçlıyor. Geleneksel, monoton öğrenme yöntemleri yerine, eğitim içeriğinin etkileşimli ve eğlenceli unsurlarla bütünleştirilmesi bilginin kalıcılığını artırmaktadır.

Bu tez çalışmasında, küçük çocukların (6-12 yaş) robotik kodlama öğrenmelerini desteklemek amacıyla eğitsel bir oyun geliştirilmiştir. Çalışma, günlük hayatta önem kazanan ve gelecekte daha da genişlemesi beklenen bir alan olan robotik üzerine odaklanmıştır. Amaç, çocukların robotik kodlama konusundaki farkındalığını artırmak ve temel programlama kavramlarını tanıtmak, çocukların oyunla etkileşime girerek kodlama, problem çözme, algoritmik düşünme ve analitik muhakeme gibi temel becerilerini geliştirmeleri için bir platform oluşturmaktır. Bu amaçla tasarlanan etkileşimli öğrenme ortamında, çocukların kendi robotlarını tasarlayıp, kodlama ilkelerini uygulamalı deneyim yoluyla anlamalarına olanak tanıyan işlevler oluşturulmuştur. Çalışmada ayrıca benzer eğitim uygulamaları incelenmiş ve robotik kodlama öğretimindeki etkinlikleri analiz edilmiştir. Yapılan oyun, 12 katılımcı tarafından test edilmiş ve oyun hakkındaki görüşleri değerlendirilmiştir.

Yapılan çalışmalar ve araştırmalar sonucunda, oyunlaştırmanın çocukların öğrenme süreçlerini olumlu yönde etkilediğini gözlemlenmiştir. Geliştirilen oyun, eğitim içeriğini eğlenceyle başarılı bir şekilde birleştirerek robotik kodlamayı küçük yaşlardaki öğrenciler için daha erişilebilir ve ilgi çekici hale getirdi. Bu yaklaşım, STEM alanlarına erken yaşta ilgi duyulmasına ve bilişsel becerilerin geliştirilmesine katkıda bulunmuştur. Gelecekteki çalışmalar, robotik kodlama eğitiminde oyun tabanlı öğrenmenin daha fazla geliştirilmesini ve daha geniş uygulamalarını araştırabilir.

Anahtar Kelimeler: Oyun, Robotik Kodlama, Çocuklar için eğitim, Scratch, Kodlama, Eğitsel Oyun

ABSTRACT

M.Sc. Thesis

Esma ÇELİKTEN

Manisa Celal Bayar University
Graduate School of Education
Department of Computer Engineering

Supervisor:
Doç. Dr. Birim BALCI

As technological advancements continue to evolve, access to new learning methods has become increasingly diverse. Learning is no longer confined to books or newspapers but extends to various digital platforms. These platforms aim to make learning more accessible, engaging, and memorable. Instead of traditional, monotonous learning methods, integrating educational content with interactive and entertaining elements enhances knowledge retention.

In this thesis, an educational game was developed to support young children (6-12 years old) to learn robotic coding. The study focused on robotics, an area that is important in everyday life and is expected to expand further in the future. The aim is to raise children's awareness of robotic coding and introduce basic programming concepts, create a platform for children to interact with the game and develop their basic skills such as coding, problem solving, algorithmic thinking and analytical reasoning. In an interactive learning environment designed for this purpose, functions have been created that allow children to design their own robots and understand coding principles through hands-on experience. The study also examined similar educational practices and analyzed their effectiveness in teaching robotic coding. The game was tested by 12 participants and their views on the game were evaluated.

As a result of studies and researches, it has been observed that gamification positively affects children's learning processes. The developed game successfully combined educational content with entertainment, making robotic coding more accessible and engaging for young learners. This approach has contributed to an early interest in STEM fields and the development of cognitive skills. Future studies may explore further development and wider applications of game-based learning in robotic coding education.

Keywords: Game, Robotic Coding, Education in Children, Scratch, Coding, Children, Educational Game

PREFACE AND ACKNOWLEDGEMENTS

Bu çalışmanın hazırlanmasında değerli yönlendirmeleriyle beni yönlendiren, desteğini esirgemeden bu yolda devam etmemi ve tamamlamamı sağlayan değerli danışman hocam Doç. Dr. Birim Balcı'ya içtenlikle teşekkür ederim.

Tüm çalışma sürecinde ve hayatımda bana her türlü moral ve motivasyonu sağlayan aileme, özellikle babam Mehmet Çelikten ve annem Selvihan Çelikten'e sonsuz şükranlarımı sunarım.

Ayrıca LRwtG oyununun değerlendirilmesinde emeği geçen tüm katılımcılara, anket çalışması için zaman ayıran değerli eğitimcilere ve tez savunmamda bana destek olan ve çalışmamı değerlendiren değerli jüri üyeleri Birol Çiloğlugil ve Zeynep Çipiloğlu'na teşekkür ederim.

Esma Çelikten

Manisa, 2025

INDEX OF SYMBOLS AND ABBREVIATIONS

4C	Critical Thinking, Communication, Collaboration, Creativity
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
LRwtG	Learning Robotics with the Game
STEM	Science, Technology, Engineering, Mathematics

INDEX OF FIGURES

Page

INDEX OF TABLES

CONTENTS

SECTION THREE

RESEARCH FINDINGS AND DISCUSSION

INTRODUCTION

Technological advancements have gained significant importance in today's world. We are constantly aware of and utilizing technological developments and achievements in all areas and at every level. Moreover, we are continuously discovering new technologies every day. One of these technological endeavors is the development of robots, which have begun to take their place in our daily lives and workplaces. The presence of robots in our daily lives and workplaces is seen as making tasks easier and bringing convenience to humans. With the use of robotics, even difficult tasks can be made simpler (Çengelci et al. 2005; Xiao et. al, 2022; Silvera-Tawil, 2024; Yılmaz, 2018).

The basis of robot design and production lies in robotic programming, which has given rise to new robot designs and various robotic endeavors. Robotics education serves as a catalyst for cultivating interdisciplinary collaboration and innovation by integrating principles from diverse disciplines such as computer science, engineering, mathematics, and design (Yılmaz, 2023; Oliver, 2018; Karataş, 2021). The ability to code robots is becoming an important skill in today's technology-driven world, and teaching young children the basics of robotic coding, giving them STEM (Science, Technology, Engineering, Mathematics) can provide a valuable foundation for future careers. There are many applications and designs put into use for this (Canbeldek, 2020; Yılmaz, 2023; Oliver, 2018).

Robotics programming is becoming increasingly significant due to the rapid advancements in technology in today's world. This discipline, which serves as a fundamental building block in the design and production of robots, has a significant impact not only in industrial applications but also in the field of education. In particular, teaching robotics programming to children is crucial for the development of future technology leaders (Yılmaz, 2023; Şahin,2021). The integration of robotics education into children's learning experiences encompasses broader socio-economic, interdisciplinary, and ethical dimensions, transcending mere technical proficiency. By nurturing a generation of robotics-literate individuals, educators not only prepared learners for future jobs but also empowered them to actively shape technological advancements (Çetin et al.2018). The integration of technology-based training (robotic coding, STEM, etc.), science, technology, engineering and mathematics that can be given to children, bridges their content and real-life linking will provide experience for young children (Çetin et al.2018; Breiner et al., 2012; Ejiwale, 2013). Teaching robotic coding to children helps them develop future work and life skills while also increasing their interest and understanding of technology. In this way, it contributes to the education of qualified individuals who will contribute to the technological transformation process of societies (Çetin et al. 2018; Yılmaz, 2023; Kanmaz, 2023; Canbeldek, 2020).

Robotics coding helps to develop children's critical thinking, problem-solving skills, and creativity. Through programming-based education, children gain the ability to break down complex problems into smaller parts and develop step-by-step problem-solving skills. Additionally, it encourages making mistakes and learning from them, thereby promoting important character traits such as resilience and patience. Furthermore, robotics coding education has been shown to improve children's academic achievement, support language development, reduce distractions, and support creativity and imagination in children (Kanmaz, 2023; Canbeldek, 2020). By encouraging collaboration among young students from different backgrounds and perspectives, robotics education nurtures a culture of teamwork and cooperation, which is essential for addressing complex societal challenges and driving meaningful change (Yılmaz, 2023; Yılmaz et al. 2023; Şahin, 2021).

Early childhood, on the other hand, is a critical period in terms of development of child and if a quality education is provided, it has positive effects on the development of the child, academic and social aspects (Burchinal et al.2010; Canbeldek, 2020; Yılmaz et al. 2023). In today's world, early childhood children who are stated to be Generation Z are raised directly in technology. For this, more computer programming tools are being created and used in early childhood education classes than ever before for young children

(Strawhacker et al. 2018). If robotics coding education is given to children at a young age, it also has many benefits.

Firstly, this education improves a child's ability to see a task through to completion. By identifying problems, breaking them down step by step, and reaching a solution, children's confidence is boosted. Secondly, robotic coding education develops problem-solving skills. Children learn to approach problems logically and create solutions. This process also enhances children's creativity and imagination, as they turn the ideas they have conjured up during coding into reality. (Kanmaz, 2023; Karataş, 2021; Yılmaz et al. 2023; Şahin et al. 2024; Yaman, 2023).

Additionally, robotic coding training increases interest in STEM (Science, Technology, Engineering, Mathematics) at a young age and can be an inspiration for a career in these fields. In addition, with the digitalization of the education system, robotics coding courses enable students to grow intimately with technology and deepen their understanding of technology. (Şahin, 2021; Barradas et al. 2019; Yılmaz et al. 2023).

In this thesis, it is aimed to design an educational game so that robotic coding education for children can be presented in an interactive environment. Robotic coding education studies for children are examined in detail and an educational game design is planned. The design will be aimed at children and will be simple and understandable. In the educational game environment to be designed, the child will be able to create his/her own robot. It is aimed for children to able to learn by having fun while designing his/her own toy. When children play, they will also learn the basics of robotic coding and they will make their robot to perform different functions in the game. They will be able to create their own design/designs (more than one is possible) and assign specific functions to these designs. So, by the help of the designed educational game environment, children can learn by having fun.

SECTION ONE
RELATED STUDIES

In this thesis study, we will create a game design to give robotic coding knowledge to individuals in early childhood. So here, some definitions (like robot, robotics, games and gamification) related to the topic are given and the information is provided about the literature.

1.1 Related Terms

A *robot* is a machine designed to carry out one or more activities quickly and accurately on its own. It varies based on the materials used and the intended usage. There are several applications, kinds, and areas. A large number of robots are used in various fields including health, construction, industry, education and production research. As new ones are created every day, the applications and types of robots are growing. (Çengelci et al. 2005; Xiao et al. 2022; Silvera-Tawil, 2024; Yılmaz, 2018; Kocaçil, 2020). These are mechanical constructions made possible by a variety of instruments and sensors. They can have a wide range of features, including wheels, lighting, arm and leg structures, and more. Depending on how they are to be used, their exterior look changes.

Robotic coding involves creating a set of instructions that tells a robot how to move and perform tasks. The process begins with defining what you want the robot to do, then programming it with the necessary code, and finally uploading this code to the robot’s memory so it can execute the tasks. There are many different coding tools available, and depending on the complexity of the task, various programming languages can be used to suit different levels of programming (Kocaçil, 2020). Robotic coding education improves young children's programming skills while helping them improve in areas critical for academic achievement such as motivation, school attendance, peer collaboration, and task engagement (Bugmann et al. 2018). Also, it contributes to the personal development of students, provides programming skills and encourages cooperation (Avci et al. 2021).

In many studies, *game* is described as “a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.” (Salen et al. 2003; Özdemir, 2023). Applications used to have fun and spend time are also used to teach things nowadays.

Computer games are applications that people of all ages now play and interact with at least once. For the first time in history, a two dimensional "Spacewar" game was released in 1967 under the name of computer game. 3-dimensional states have been created together with the later developing technology. Now, real-life similar environments are offered to users with their creative and realistic scenarios. They are games created in computer environments where imaginary scenarios are created and made available in various environments. It usually continues to play until a goal and goal is achieved (Korimov, 2021; Kılıç, 2018). Nowadays, information transfer has started using games. There are various studies on this. When the use of different methods and the benefits of information transfer were examined in the studies, repetitions and entertainment content in the games caused positive effects in learning (Tasdemir et al.,2016; Alan, D. 2017; Tekin et al., 2020).

Educational games are the games that aim to help students in learning. Also they can be described as the games that aim to increase participation in trainings and increase learning potential by providing motivation and entertainment for young children (Ibrahim et al. 2009). A good methodology and guidelines are required to design an effective learning game that balances traditional learning content with entertainment criteria (Ibrahim et al. 2009). It is important to provide well-designed games and adequate resources for effective education. In order to do this, very detailed research is required, and the educational materials should be suitable for child psychology and the age groups.

On the other hand, another term *gamification* which is seen as a positive addition to educational tools (Foldager et al.2017), can be described as the application of game design elements in non-game contexts. And gamification can be a significant factor in enhancing the learning experience in robotic coding (Andrias et al., 2021; Özdemir, 2023). The adaptive gamification design proposed, which aligns user/player types with game elements, could be particularly relevant for creating a tailored learning environment in robotic coding. But the effectiveness of gamification in education is not uniform, and its design must be approached with a nuanced understanding of the target audience and the desired outcomes. While gamification has the potential to enhance the learning experience in robotic coding, its success depends on the thoughtful integration of game elements that align with the learners' preferences and the learning objectives. (Foldager et al., 2017; Puritat, 2019). Interestingly, while gamification is generally seen as a positive addition to educational tools, suggest that the difference in engagement between gamified and non-gamified versions of an application may be minimal (Foldager et al.2017). This could imply that the mere presence of gamification is not a guarantee of increased engagement, and careful consideration must be given to the implementation of game elements. Moreover, it indicates that the impact of specific game elements on educational performance can vary, suggesting that not all gamified elements are equally effective in all contexts.

While gamification has the potential to enhance the learning experience in robotic coding, its success depends on the thoughtful integration of game elements that align with the learners' preferences and the learning objectives. The effectiveness of gamification in education is not uniform, and its design must be approached with a nuanced understanding of the target audience and the desired outcomes (Foldager et al., 2017; Puritat, 2019).

Moreover, when we consider robotic coding and educational games together, using games in robotic coding education provides positive contributions to children's learning process. Due to the problems such as lack of time, lack of materials and lack of practice have also been observed in robotic coding education (Avci et al. 2021), we may say that robotic coding education through games improves children's programming skills, leads to positive results in areas such as motivation, cooperation, and participation, offers a fun and engaging learning experience while improving students' academic performance (Thanyaphongphat et al., 2023). Integrating games into robotics education can pose technical challenges, such as compatibility issues and the need for adequate resources such as hardware and software.

1.2 Case Studies in the Current Market

When literature research is done, many studies are encountered. As a result of the researches, it is seen that, there are various studies that have been created and made available for the purpose of developing coding knowledge and teaching while entertaining. These studies are mostly game designs that use color images to advance knowledge or teach things from scratch. We will be looking at some of these designs.

1.2.1 Robo-code

Robo-code game, whose start page is shown in Figure 1, is defined as follows. “Robo-code is a programming game, where the goal is to develop a robot battle tank to battle against other tanks in Java. The robot battles are running in real-time and on-screen.” (Robocode, 2023, October 24). Users are asked to use the java or .Net coding language.

Figure 1: Robo-code game information and download link screenshot (Robocode, 2024).

Robo-code has a design structure similar to the minefield game. It welcomes the user with a playground consisting of a large square and many side-by-side small squares. The aim of the game is to be the first of the tanks that are shooting at each other. For this, users use tank silhouettes. They can use predefined tanks or design their own. For this, they are asked to use the java or .Net coding language. As a result of the operations made from the coding screen, it is ensured that the functions such as firing, moving and appearance are defined with the tank. Virtual space of the game that is designed for battle and the virtual representation of tanks can be seen in Figure 2.

Figure 2: The Robo-code game screen features the visual representation of tanks, and the virtual space designated for battle (SourceForge, 2024).

The guide offers a comprehensive overview of the procedures involved in playing and constructing the tank, with a clear and systematic presentation of fundamental concepts. It is necessary to have a basic understanding of Java in order to utilize the application. It is necessary to download both the application and the Java file during the initial installation phase. It is possible to play on a computer, although the image quality is poor and tanks cannot be displayed with clarity. Players have the option of re-coding and re-encoding in order to emerge victorious in a battle. The tank belonging to the player who writes the most successful code sequence becomes the first. This game platform is designed for those who enjoy playing war games and for users who wish to enhance their coding abilities or learn a coding language (Kılınç, 2017).

1.2.2 Code-Monkey

Code-monkey is a program based on the movement of the monkey on the screen using Python or Coffee Script programming languages. He wants to solve various puzzles in order to ensure that the monkey reaches the bananas. It has a 2-dimensional image. The game character monkey is asked to reach the banana. The game visual is displayed as in Figure 3. As teachers and parents, it is desirable to create user entries. The user base consists of children between the ages of 5 -14 and teachers and families who provide supervision of their use (CodeMonkey Studios Inc).

Figure 3: Code-monkey game page screenshot (Kılınç, 2017).

1.2.3 Coding Game

The application, which offers more than 25 programming language options, offers activities such as puzzles, game, virtual escape rooms. Along with the Login function, it provides competition with other users. The first screen opened contains guide applications. After the selected coding language, it enables us to win in the game by offering sample code blocks. On the side screen of the game, the code block is opened, and it is desirable to follow all the steps individually. The goal is to have fun while developing coding skills.

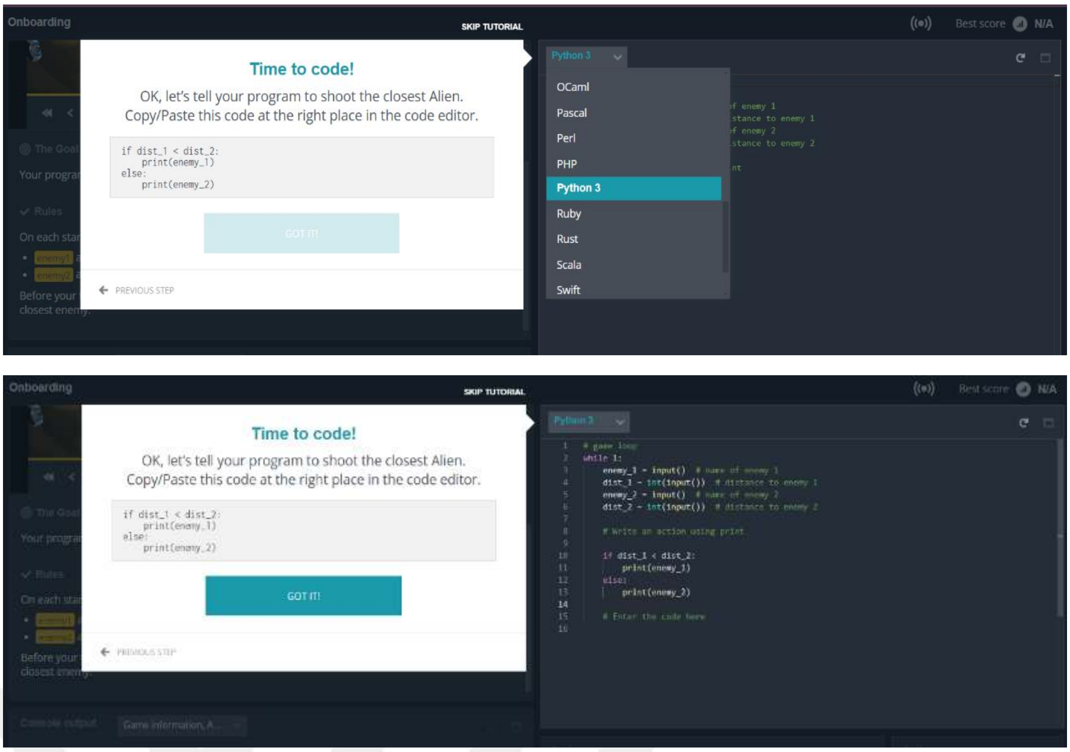

Figure 4: The Coding Game interface a- Section page for commencement guide, tools, programming language. b- Screen after the code copy-paste operation. (*CodinGame, 2024)*

As can be seen in Figure 4a, the options to be used in the game (the language to be learnt) are presented to the user. The user is asked to choose among them. As a result of the language selection, how to play and how to code in the game is shown in Figure 4b.

The Coding Game interface presents the user with a series of options, including a start guide, tools, and a screen where the programming language can be selected. Figure 4 shows a screenshot that offers programming languages to the user to be used. User can select the programming languages like C, C#, C++, D, Go, Perl, Python, etc.

Moreover, the Coding Game provides a comprehensive introduction to the fundamental aspects of the programming language, accompanied by a user manual, an array of tools, and a detailed display of the language's usage. Figure 4 (a and b) shows the secreen that enbales user to select a commencement guide, and to view the introductory section of the example code and commence the game.

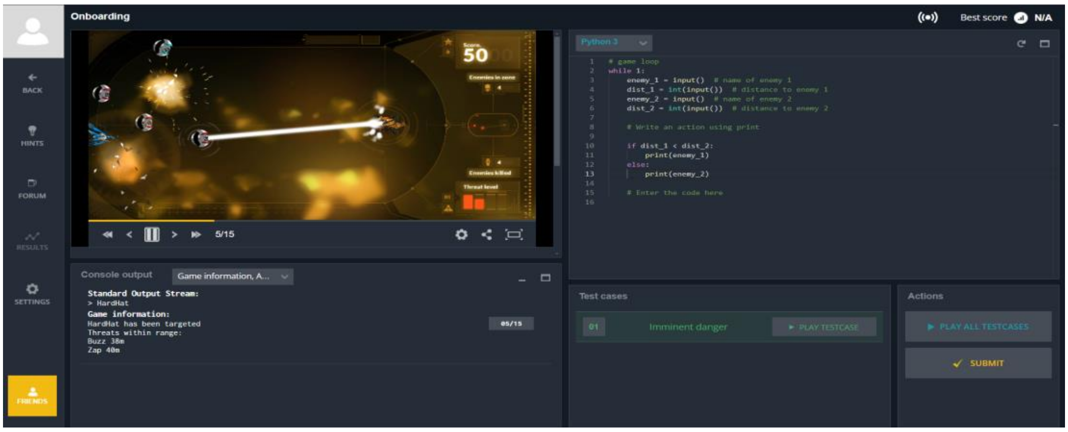

Figure 5: The Coding Game interface. (CodinGame, 2024)

Coding Game interface as shown in Figure 5, the user starts the game screen after completing the desired coding steps. The user starts shooting at the spaceships on the game screen. As a result of the firing, the war starts.

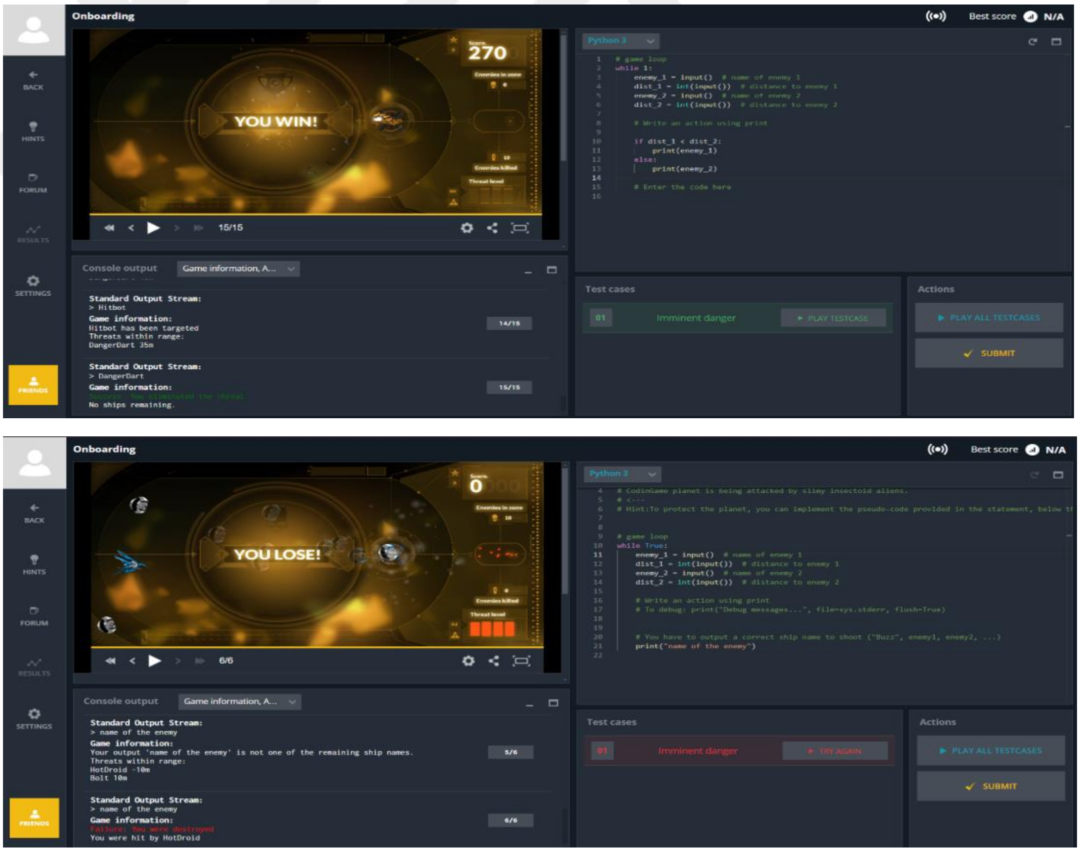

Figure 6: The game result screen. (CodinGame, 2024)

Figure 5 and Figure 6 show the coding stages of the game. Once the preliminary stage of coding has been completed, the subsequent stages are initiated on the Coding Game screen. The battle game starts, and the spaceships shoot at each other. The game continues until all spaceships are destroyed and only one remains. When the game is over, when there is only one spaceship left, the text ‘You won’ or ‘You lost’ appears on the game screen as shown in Figure 6.

“One of the best things about Coding Game is that it is installed on a legendary base that allows you to participate in international coding competitions, just as you can play with your friends or coworkers.” (Kılınç, 2017).

1.2.4 Scratch Labyrinth Game

The Scratch application is a software tool that allows users to create projects through block coding. In this example (Figure 7), the information provided relates to the maze game. The app offers two separate sets of information: one for the mechanics of the game and the other for guidance on how to play. The Scratch app offers the opportunity to create similar maze games, thus facilitating the diversification of the knowledge base. The Scratch app offers many opportunities for users. The capacity of the user's imagination, whether child or adult, is involved in this process. The way users (children and adults) conceptualize a game, animation, etc., allows them to move in this direction.

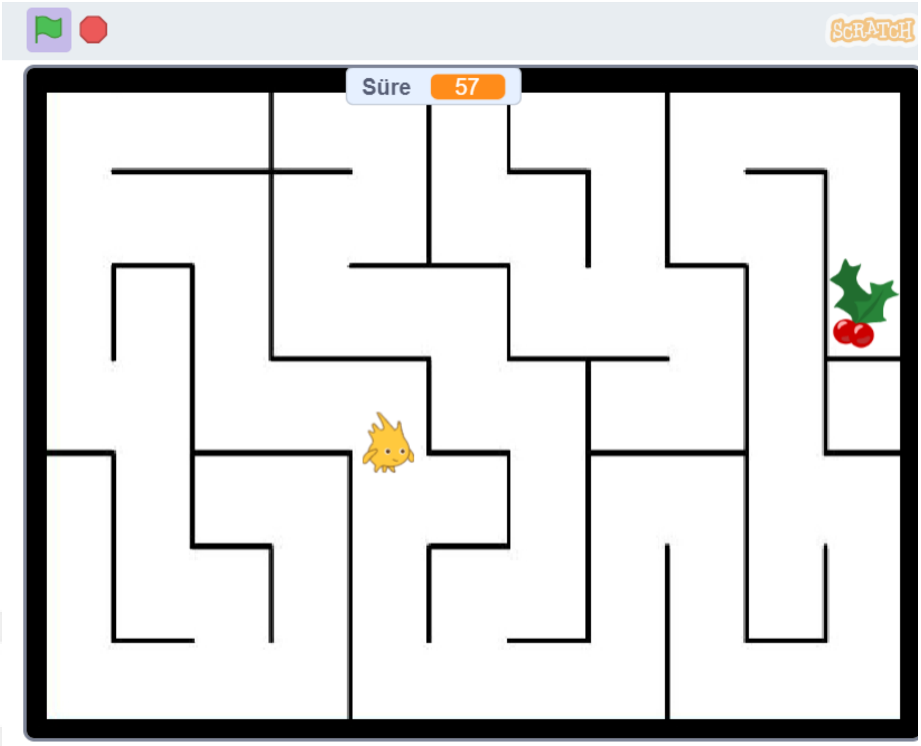

Figure 7: Scratch labyrinth game display (Özdemir, 2023).

In the example, readers are offered a brief overview of the steps involved in creating puppets, including instructions on how to move them, how to add a background, and how to make a dummy. In this game, instead of learning to code yourself, the steps involved in creating a game using one of the coding tools are summarized (Özdemir, 2023). Figure 7 is a game-play screen shot that shows how the game is played.

1.2.5 Blockly Games

Blockly Games is an educational game series that teaches programming. It is designed for children who do not have prior experience in computer programming (Blockly Games, 2024). Blockly Games is based on teaching coding concepts instead of actual programming language to children who have no previous coding experience. With free coding games for kids, the platform allows kids to have fun while helping them grasp the logic before learning robotics and coding (Arslan, 2023). At the end of these games, you are ready to use traditional text-based languages (Blockly Games, 2024).

The main page of the Blockly Game is seen in Figure 8. The descriptions of eight icons that are seen in the figure are explained below:

•*Puzzle:* This activity provides a basic introduction to the shapes of Blockly and demonstrates how different pieces interconnect, offering a visual approach to learning programming structures.
•*Labyrinth (Maze):* This task introduces the concepts of loops and conditional statements. It begins with simple challenges, progressively increasing in difficulty to enhance the learner's understanding of control flow.
•*Bird Conditional:* This exercise delves deeply into conditional structures, focusing on how control flow is influenced by multiple, increasingly complex conditions.
•*Tortoise Loops:* Here, learners explore nested loops as they work to construct visual representations, reinforcing their understanding of loop structures in programming.
•*Film:* This module introduces the use of mathematical equations in programming. It demonstrates how mathematical concepts are applied to create a film, providing a practical context for the use of mathematics in coding.
•*Music:* In this activity, learners are introduced to the concept of functions in programming. They are required to apply functions to create a musical composition, highlighting the utility of functions in problem-solving.
•*Pond Tutor:* This section transitions to text-based programming, alternating between graphical blocks and real JavaScript code within a text editor. This allows learners to practice both visual and textual forms of programming.
•*Pond:* An open-ended challenge, Pond encourages learners to program the "smartest" duck using either blocks or JavaScript. This competition encourages creative problem-solving through both visual and text-based coding techniques.

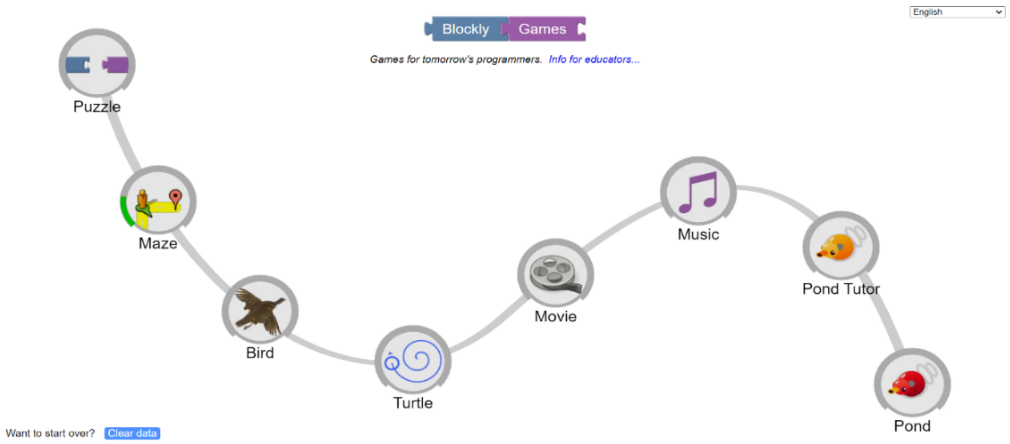

Figure 8: Blockly games main page display (Blockly Games, 2024).

Figure 9 shows the Labyrinth (maze) game of the Blocly Games application. On the Labyrinth (over) game tab, users are asked to present and complete a variety of challenges. The target is tried to be reached by using commands such as movement, right turn, left turn, repeat. Increasingly difficult transactions are presented at each level. After the completed difficulties, the Java script code is presented to the users (see Figure 10).

Figure 9: Blockly Games: Labyrinth (maze) game display (Blockly Games, 2024).

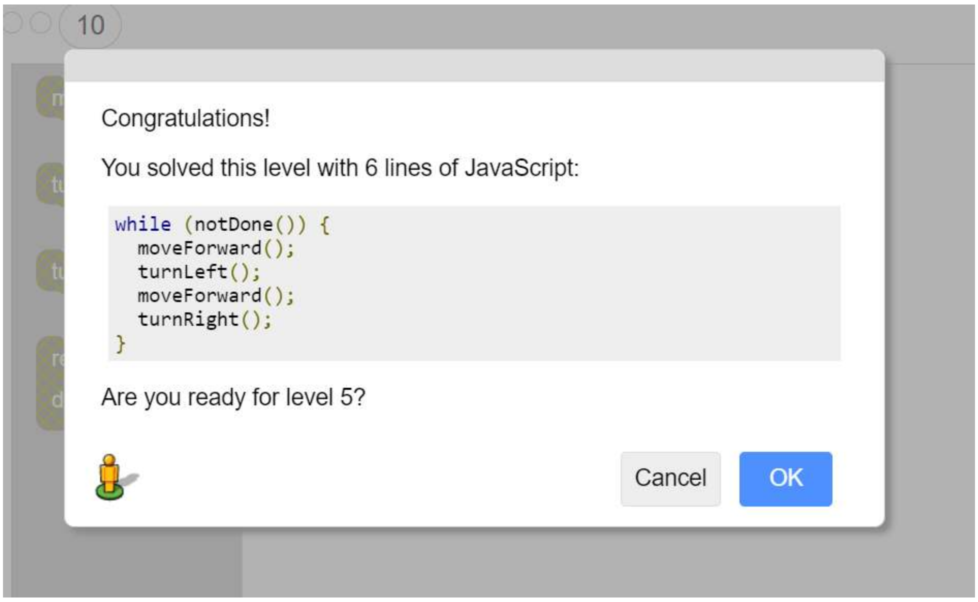

Figure 10: Blockly Games: Labyrinth (maze) game congratulation display (Blockly Games, 2024).

1.2.6 Tynker

Tynker (see Figure 11) is actually a platform where children can play coding games for free and prepare projects more than a game. There is also a panel where parents can follow coding activities and the development of their children on the platform, which can be registered free of charge (Arslan, 2023).

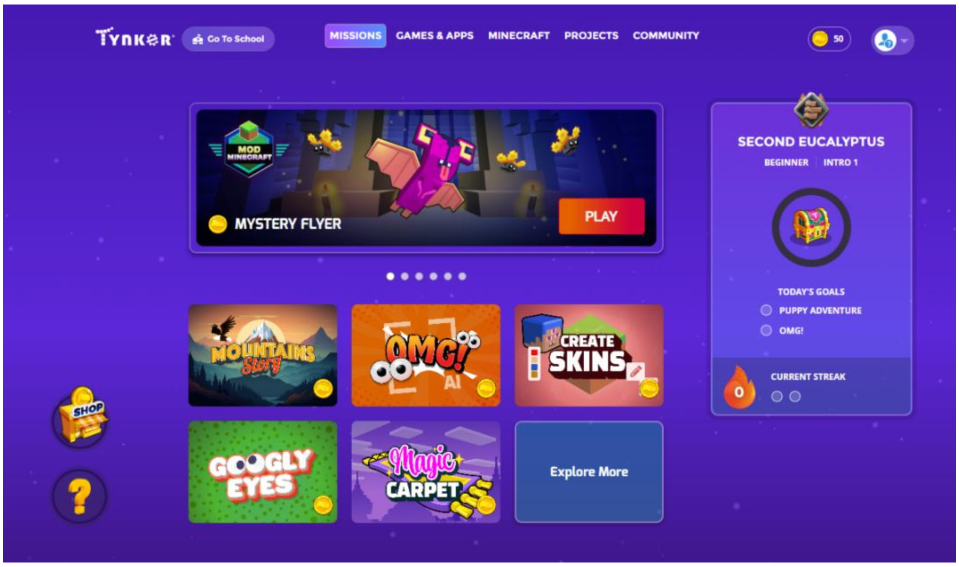

Figure 11: Game options screen shown to children after user login (Tynker, 2024)

The Tynker platform defines its mission as; “Tynker’s mission is to provide every child with a solid foundation in computer science, programming, and critical thinking skills to prepare them to become better architects of their future world. Our goal is to ensure that, by 2033, every high school graduate has been exposed to some formal coding education. (Tynker 2024).

A carefully designed program consisting of 300 learning modules at six certification level to transform the child from beginners to real-world programmers, also Minecraft Modding, Game Design, Creative Coding, Introduction to Artificial Intelligence, etc. It includes 30 gamified electives to apply coding knowledge to areas such as Data Science and more. It aims to reach every child by keeping the number of diversity high. It is a colorful platform designed to lay the foundations for the upbringing of future programmers (Tynker 2024).

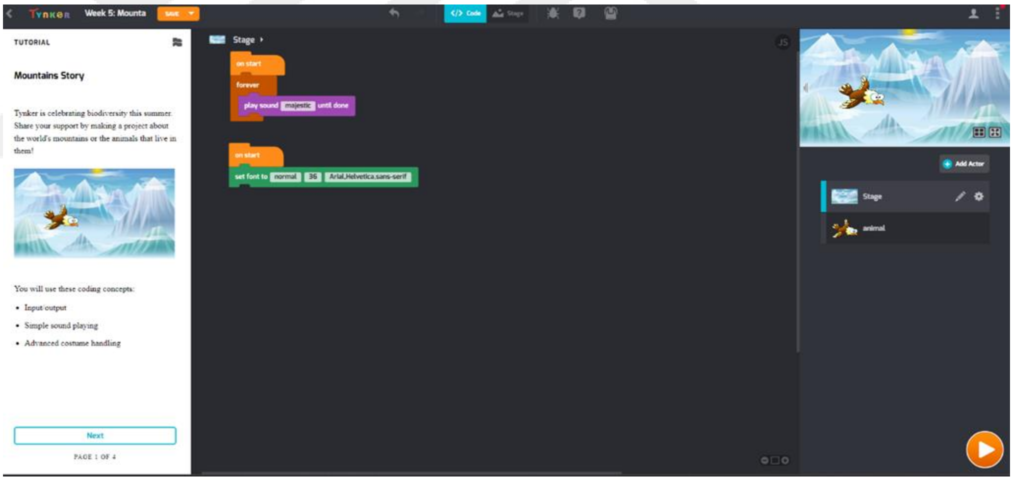

Figure 12: Mountains Story code screen display (Tynker, 2024)

The Tynker application offers users (children) a multitude of opportunities for interaction. The app offers a variety of interactions, including artificial intelligence teaching, web page design, coding skills, robotic coding skills, and data analysis skills. Users (children) are instructed in coding languages such as Java, JavaScript, Python, Micro Python, HTML, and CSS. The application provides a range of interactions (game content) for these coding languages. Its colorful interface and extensive content are designed to capture the attention of users.

It attracts the attention of children with its colorful contents and diversity. It diversifies educational content using today's popular games. In Figure 12 and Figure 13, Mountains Story code screen and Space Quest game screen are shown respectively.

Figure 13: Space Quest game screen display (Tynker, 2024).

1.2.7 Code Combat

Code Combat, a code writing game for kids, is a game of creating solutions by writing code for puzzles and problems. The game has categories such as computer and web development. It is planned to develop at the coding learning level with each level passed. (Arslan, 2023).

Code Combat is a game for the teaching of JavaScript, Python, and the Lua language. This online game is a game that improves strategy determination. The game has the option to be available in many languages. The motto of the game is to learn to code by playing games. The use of children in the development of games has been targeted, making it suitable for the use of users of all ages. Code Combat is an online gaming platform (Aytekin at al., 2018).

Figure 14: Code Combat, game home screen that shows game selection (CodeCombat, 2024)

It is an application consisting of tasks to be completed and coding and coding tips for these tasks. Figure 14 shows the road map to be followed in the game. Each step on this map is required to be completed. Figure 15 is only one of the game steps to be completed in Figure 14. Here too, users are given a new road map. They are asked to complete the steps on this road map. Users are given hints to complete each game step, as shown in Figure 16. The game steps are continued as the steps on the hint screen are completed. After the game phase is completed and the diamonds in the target are collected, the game phase is completed.

Figure 15: Code Combat game Kithgard Dungeon game level map (CodeCombat, 2024).

Figure 16: The screen of Code combat game that show the screen. (CodeCombat, 2024).

1.2.8 Code.org

Code.org is an educational innovation non-profit organization dedicated to the vision that every student at every school has the opportunity to learn computer science and artificial intelligence. Since 2013 it has continued to serve. The aim of the application is to teach children to code through play. Through this platform, it is aimed for children to acquire basic algorithmic skills. Code.org is supported by generous donors, including Microsoft, Amazon, Google, and others (Code.org 2024; Aytekin at al., 2018).

With Code.org, children are taught not only to write code, but also to create cycles. They are asked to contribute by making small additions to the work on the system by improving problem-solving abilities.

The cartoon character was created, and the children were asked to use these characters to go back and forth with block commands and direct them using commands that rotate left and right. There are many characters on this platform. In this way, children can learn to code on the application by both having fun and learning (Aytekin at al., 2018).

Code.org is a platform where children can prepare various projects, examine the codes of prepared projects, access coding games and practice coding. On this platform, children will be able to learn and practice coding knowledge through various games or projects and share them on the platform.

They will also be able to develop a different perspective by examining the codes of the projects prepared by other children (Arslan, 2023).

The game wants to complete the steps on the colorful game screen provided by various cartoon characters. Each step is completed by applying the operations on the coding screen. The process is repeated until the character reaches the target. In Figure 19 you can see the plan map of the game steps.

After each operation is completed, the difficulty level is increased step by step and the game continues to be played.

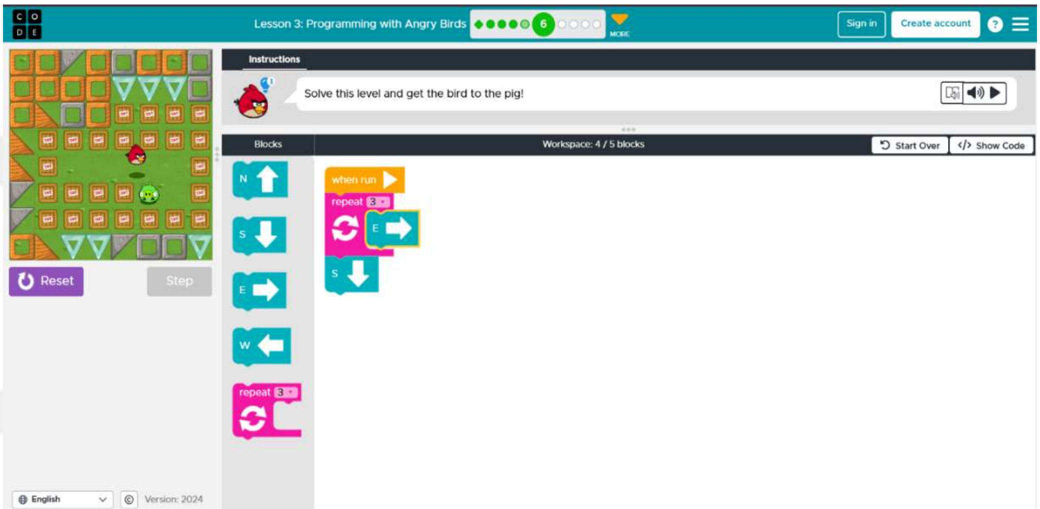

Figure 17: Code.org game screen code steps (Code.org, 2024-a)

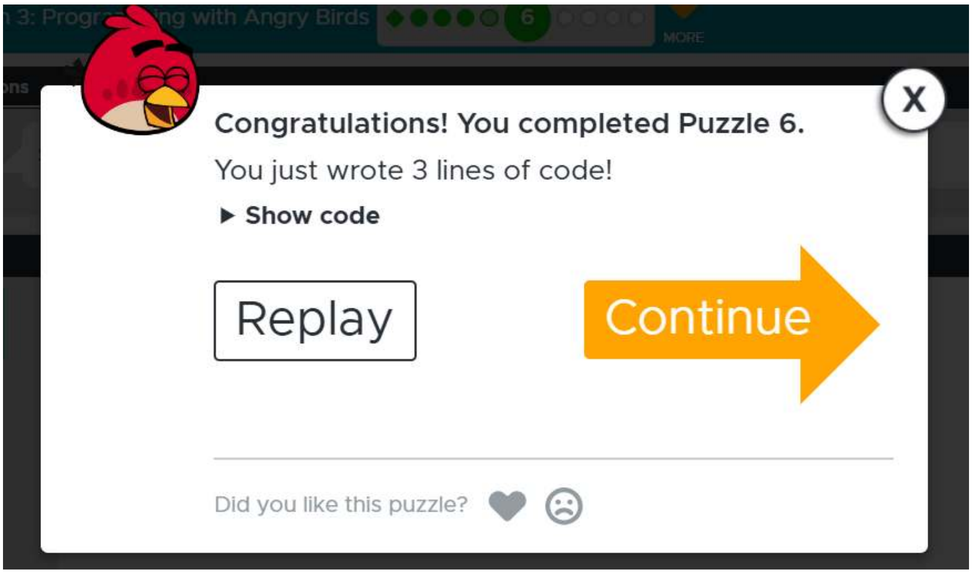

Figure 18: An example screen that shows the user successfully completed the puzzle 6. (Code.org, 2024-a)

Code.org provides users (children) with a lot of interaction. Some of these interactions include; Artificial intelligence teaching, web page design, coding skills, robotic coding skills, data analysis skills. In the application, users (children) are taught coding skills using coding languages such as; Javascript, Html and Css.

Plan map with the stages planned and taught in the game (Figure 19). As the game steps are completed, which programming stage is taught and what game level to teach can be viewed through this map. Each completed step is indicated by green color.

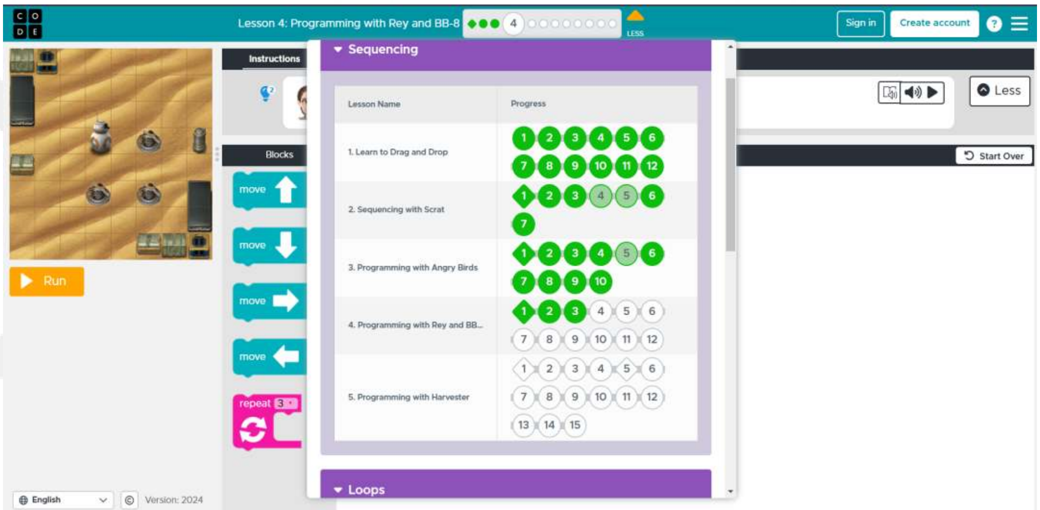

Figure 19: Plan map with the stages planned and taught in the game (Code.org, 2024-a)

1.2.9 Crunchzilla (Code Monster)

Code Monster from Crunchzilla is an interactive tutorial for kids that focuses on action. Code changes give immediately visible results. Projects start with simple boxes and colors (Figure 20), quickly progress to exciting experiments with simple animation and fractals. Important programming concepts such as variables, loops, conditions, expressions and functions are presented as examples.

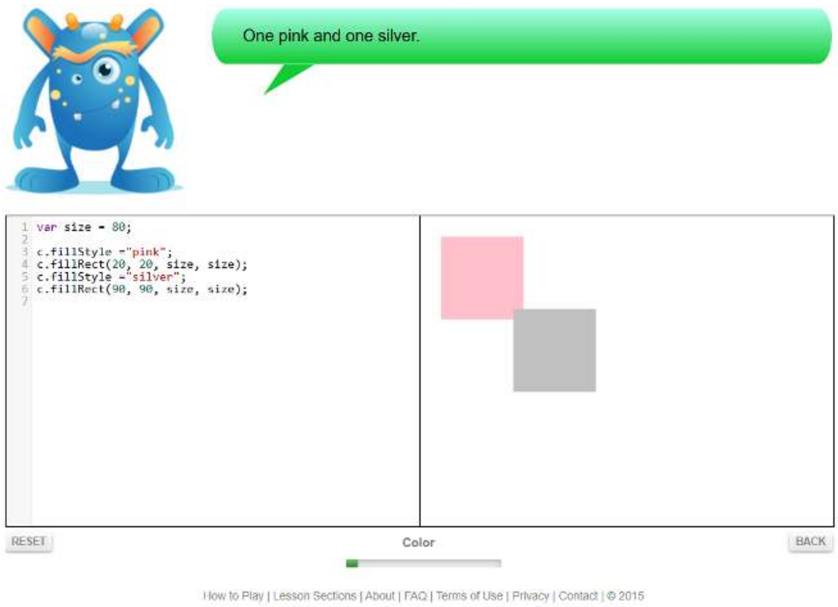

Figure 20: Code Monster image and javascript code screen from Crunchzilla (*Crunchzilla, 2024)*

Code Monster is a gentle and fun introduction to programming concepts. This is the first step to learning programming. It is not intended to teach all computer science and programming. It offers children the opportunity to learn by exploring. Access can be provided free of charge (Crunchzilla, 2024). Code Monster is the screen that makes it easy to switch between lessons sections and lets us see what steps are in place (Figure 21).

Figure 21: Code Monster is the screen (Crunchzilla, 2024)

1.2.10 Code Karts

It is a car racing game designed for young children. It is aimed to bring the car to the finish line by defining the direction keys respectively. As each level is completed, the road difficulty that the car must pass is increasing. The option of playing in the form of a single car or two cars is offered in the game. It is necessary to download to a computer, tablet or phone to play the game. Colorful designs were used in the game to attract the attention of children. In order to prevent boredom and to ensure sustainability, different difficulties have been created in the game as the level of the game is completed (Code Karts, 2023; Arslan, 2023).

Code card game offers options to play with one car or two cars. This shows that two people can play the game at the same time. Figure 22 shows the game screen played with one car, i.e. one user. Figure 23 shows the game screen with two cars, which allows two users to play the game at the same time.

Figure 22: Game screen played with a car. (GameLoop, 2023)

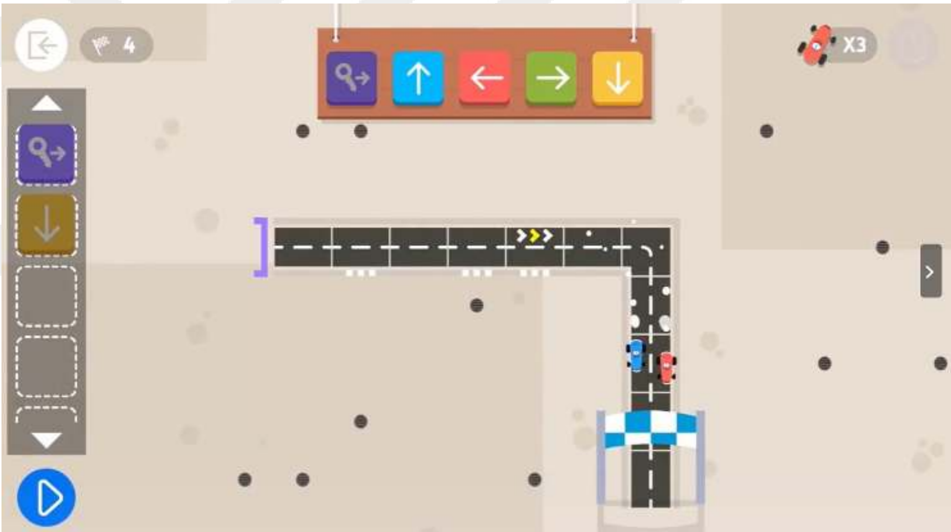

Figure 23: Code Karts game screen played with two cars. (GameLoop, 2023)

1.2.11 Kodable

Kodable Basics introduces basic coding concepts and familiarizes students with Kodable game mechanics. Kodable Basics includes short tutorials that students can complete in 5 minutes or less (Kodable-a. 2024). Each Kodable course is a collection of game levels and videos on a specific topic. In the game, each route is symbolized by a planet (Figure 24). It allows students to travel around a map of the galaxy, allowing all courses to be explored (Mattingly, 2024).

Figure 24: Kodable game start screen (Kodable, 2024)

Since it is an old version, it took a long time to load the demo part. However, the steps to be implemented in the game provide simplicity. Apart from the exploration of planets and step-by-step path progression games (as shown in Figure 25), a separate pet feeding section was created where they could experience the coding stages. For the care of this pet, block coding expressions such as clean, give food, give toys were presented to the user. The user is asked to perform the pet care by sequencing the block codes.

Figure 25: Kodable game level map (Kodable, 2024)

Figure 26: Game screen asking for a sequence of directional keys to move the character (Kodable, 2024)

1.3 Some Thesis Studies on the Subject

In the academic domain of robotics education, there exists a substantial body of literature addressing the healthy, sensory and social development of children. The present studies proffers a multidisciplinary perspective, encompassing the contributions of diverse age groups, an array of technological tools and pedagogical approaches. A thorough examination of extant theses in this field has yielded significant findings, underscoring the educational potential of integrating robotic coding education with gamification and learning methodologies.

In the study by Dizman (2018), is utilized that an 8-week training course that incorporated the LEGO Mindstorms EV3 set. The programme was designed within the framework of the 4C approach (Critical Thinking, Communication, Cooperation, Creativity), enabling the participants to evaluate their programming competencies using a tangible robot. It has been observed that the scientific study of robotics, the comprehensive thinking of robotics education, and its collaborative work improve innovation and problem-solving abilities.

Egin (2019) examined a master's thesis. The findings from this study have been utilised in the instruction of 105 Information Technologies teachers operating within the context of the Kodla(Ma)nisa Project in Manisa. The study revealed a clear preference for molecular block-based and robotic writing tools, in addition to visual and game-based teaching methods.

In master's thesis, Konyaoglu (2019) examined the effects of robotic coding training on the problem-solving skills of secondary school students and their views on robotic coding activities. The study utilised a five-week single-group scanning design, employing Edison training robots and Makeblock mBot robots. The research findings indicate that a five-week robotics training programme exerts a substantial and favourable influence on problem-solving abilities.

In master's thesis, Eraytac (2019) sought to examine the impact of a block-based teaching method on the academic achievements of secondary school students in the context of robotic coding education. The study involved the implementation of robotic training, utilising the method of distribution in blocks Informatics Technologies and Software in secondary schools. The research design incorporated a semi-experimental approach, employing a post-test design to assess the effectiveness of the block-based teaching method. The study's sample size comprised 130 students. It has been observed that the experimental group, which received training based on the method of delivering block-based (mBlock), demonstrated higher academic success in comparison to the control group, which received training based on the text-based (Arduino) method. This study is significant in revealing the superiority of the block-based coding method in comparison to the text-based method, particularly for novice programmers in middle school. The pedagogical efficacy of block-based coding in facilitating the conceptualisation of abstract programming concepts is well-documented, with a concomitant positive impact on students' academic achievement. Moreover, also Alper and Şanlı's (2024) study provides support for the assertion that block-based technologies are the most popular tools, with the Scratch platform ~~b~~eing the most prominent among them.

In the study Canbeldek (2019), the mathematical development of the "Producing Children Coding and Robotic Education Program" applied to children aged 5-6 years was examined, as well as any improvement in language. The research results indicated significant increases in experimental groups in terms of strategies employed by individuals, as well as in language and creativity scores. The study demonstrated that the provision of coding and robotics education in early childhood is conducive to the cognitive, linguistic and creative development of children. And it is recommended that early childhood education teachers use coding and robotics education programmes that are compatible with their activities.

In the study by Fidan (2016), it was observed that the approach of gamification using Scratch software was presented to the student. The present study employed a qualitative research method and was conducted with a sample of 37 students. Within the context of education, gamification elements such as score tables, badges, levels and characters have been utilised. The research results of the study demonstrated that the incorporation of gamification in education led to an enhancement in student output and motivation, a more enjoyable educational experience, and an increase in academic achievement. Thesis work supports the hypothesis that educational robot activities, which are enriched with games, have a positive impact on class participation and motivation, thereby rendering the learning process more enjoyable (Kaya et al., 2020).

The study of Totan (2021) investigates the impact of Block Based Coding Education on the Attitudes of Middle School Students towards Information Processing, Thinking Skills and Coding Learning. The findings of the research which includes 38 students demonstrated a substantial positive enhancement in their self-efficacy and their attitudes towards discourse regarding information-processing thinking in the domain of block-based education.

In addition, apart from these theses, Yılmaz and Kara (2023) mentioned in their study where they made a review on the methods and findings of thesis research in Turkey that Robotic technologies have been found to support education in early studies. They also mentioned that, success in education increases academic performance, creativity and problem-solving skills, as well as STEM skills. The acquisition of robotic technologies has been demonstrated to facilitate the development of students' intelligence, critical thinking and social-emotional abilities. Furthermore, individuals who are introduced to robotic growth at an early age acquire technological literacy, improve their ability to analyse complex problems and produce solutions (Yılmaz and Kara, 2023).

Similarly, Haymana and Özalp (2020) conducted an experimental study examining the effects of robotics and coding education on the creative thinking skills of 4th-grade primary school students. Their research involved 60 students divided into experimental and control groups, with the experimental group receiving 10 weeks of robotics and coding education. The results demonstrated that students who participated in robotics and coding education developed significantly enhanced creative thinking skills compared to the control group. The researchers concluded that robotics activities create a learning environment that encourages students to think divergently, generate multiple solutions to problems, and develop their imagination. This study provides empirical evidence supporting the assertions of Yılmaz and Kara (2023) regarding the development of creativity and problem-solving skills through robotics education, particularly when implemented at the primary school level.

Moreover, as demonstrated in the research conducted by Alper and Sanli (2024) that examines 194 postgraduate theses in the National Thesis Centre Database, there is a growing body of academic studies that support the notion that robotic training exerts a positive influence on strategic points in academic achievement and problem solving. In their study, Alper and Şanlı posited that robotic education has been demonstrated to enhance academic success and problem-solving skills in feathers.

SECTION TWO
MATERIALS AND METHODS

This thesis study aims to integrate entertainment and education within the same platform, ensuring that children not only engage in an enjoyable experience but also acquire essential skills and knowledge in robotic coding. The proposed game is designed to enhance children's robotic coding abilities by allowing them to progress at their own learning pace. To achieve this, various game elements, including barriers, puzzles, and animations, were incorporated to facilitate an interactive and adaptive learning environment. The materials and methods employed in this thesis study are detailed below.

2.1 Materials

The primary material used in this thesis study is the Scratch programming platform. Scratch is widely recognized as an effective educational tool for teaching robotics coding due to its visual programming interface, which simplifies the learning process for students without prior coding experience (Nolen et al., 2018; Attila et al., 2022). This approach is particularly beneficial in educational settings, where hands-on experience in constructing and programming robot models enhances learning and develops a range of skills (Attila et al., 2022).

However, while Scratch simplifies the programming process, it has certain limitations. For instance, it may obscure the complexity of robotic systems, potentially limiting the depth of knowledge acquired by students (Pacheco et al., 2021). Additionally, Scratch's lack of support for mobile applications poses challenges in extending its educational reach (Salahli et al., 2019). Despite these, Scratch serves as a valuable tool for introducing robotics coding, particularly in educational contexts where it fosters engagement and foundational programming skills. Nevertheless, educators should remain aware of its limitations in conveying the full complexity of robotics and its current gap in mobile application support (Pacheco et al., 2021; Salahli et al., 2019).

2.2 Methods

This thesis study explored various aspects of integrating robotic coding knowledge into children's education through game design in Scratch. The questions guiding this thesis study include the followings.

- How can robotic coding knowledge be effectively transferred to children using Scratch?
- How can Scratch's visual programming environment make coding accessible and engaging for children?
- What kind of game can be developed using the Scratch app?
- What themes can be integrated into game design?
- How can robotic elements be incorporated into the game to strengthen coding skills?
- What are some examples of successful educational games that use Scratch to teach similar concepts?
- How can user interface elements be designed to enhance engagement?
- How should game design be approached from a pedagogical perspective?

A survey method was used to evaluate the effectiveness of the educational game and to collect feedback. The questionnaire was designed to collect educators' opinions on various aspects of the game, such as its design, educational content, user experience, and technical performance. The survey consists of a total of 15 Likert-type questions (with values between 1-5). A total of 12 participants were included in the survey, who were familiar with assessing the functionality of the programme and the development process.

The participants were a diverse group of professionals with relevant expertise in educational technology and programming. Five of the participants were computer engineers, while the remaining seven were instructors teaching information technology programmes at universities. The pool of participants was deliberately balanced in terms of prior experience with Scratch programming: six participants had previously used Scratch to complete small projects, while the other six had no prior experience with Scratch specifically, but were familiar with other programming languages. One participant specialised in teaching Scratch courses, while eight provided instruction in various programming languages. Additionally, two participants brought a unique psychological perspective, having been involved in various studies examining and evaluating the psychological states of children, adolescents and adults. This was valuable for assessing the game's age-appropriate design and motivational aspects.

The data collection process was carried out in several carefully planned stages. Educators were asked to interact with the game, observe its features and provide detailed answers to the survey questions. The feedback was then analysed to identify areas for improvement and to refine the game design accordingly. The questionnaire requested from the participants is as in the Table 1.

Table 1: LRwtG Game Questionnaire

Dear participant, This survey is prepared to evaluate the game called LRwtG, which was developed to teach robotic coding to children between the ages of 6-12. Your answers are important for the development of the game. Please answer the following survey questions after playing the game, taking into account the target age group. Please answer by paying attention to the fact that some questions are given on a 5-point Likert scale, where 1- very little ... 5 is very much. Thank you. Your Name Surname (Optional): 1- How would you rate the game’s potential to engage students? (1: Very Weak 2- Weak 3-Medium 4- Strong 5- Very Strong) 2- How would you rate the educational aspect of the designed game? (1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong) 3- How would you rate the game's educational content? (1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong) 4- How would you rate the level of contribution of the game to problem solving skills of the students? (1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong) 5- How would you rate the level of contribution of the game to algorithmic thinking skills of the students?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

6- How would you rate the level of contribution of the game to coding skills of the students?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

7- How would you rate whether the game is at a level that students can easily understand?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

8- How would you rate the level of fun of the game?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

9- How would you rate the game's ease of play?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

10- How would you rate the user-friendliness of the game's interface?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

11- How would you rate the game's technical features?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

12- How would you rate the game's visual design?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

13- How would you rate the overall game?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

14- At what level would you recommend this game?

(1: Very weak 2- Weak 3-Medium 4- Strong 5- Very Strong)

15- Are there any other comments or suggestions you would like to add?

…………………………………………………………………….

2.3 Development of the Game

The educational game developed in this thesis was constructed utilizing the Scratch platform. Elements of gamification, such as rewards, scores, challenges, and feedback loops, were incorporated to enhance motivation and improve learning retention. (Vygotsky, 1978; Deterding et al., 2011).

The game's design incorporated the principles of color psychology. Warm hues, such as red and orange, were utilized to evoke excitement, whereas cool tones, such as blue and green, were employed to foster concentration. The implementation of high-contrast color schemes was a deliberate design choice to enhance readability and accessibility, thereby ensuring an inclusive learning environment (Gil-Aciron, L. A. ,2024; Bevz, M. ,2023).

The game incorporates a narrative in which players guide a robot through various challenges and solve coding puzzles to advance through levels. Each level introduces a new concept in robotic coding, such as loops, conditions, and basic algorithms, thereby providing a gradual learning progression. The developed game is called Learn Robotics with the Game (LRwtG).

LRwtG greets users with a welcome page and facilitates user registration and login processes. During the login process, the user is asked for a username, password and age information. The main page of the game is shown in Figure 27. Users' login information is stored in a list within the Scratch interface. The login process is minimal, requiring only basic authentication information. It should be noted that although an attempt has been made to create a real-time database, the Scratch application only allows it to be used as a demo. The interface of the application has a user-friendly design, and the data entered is stored in generated lists.

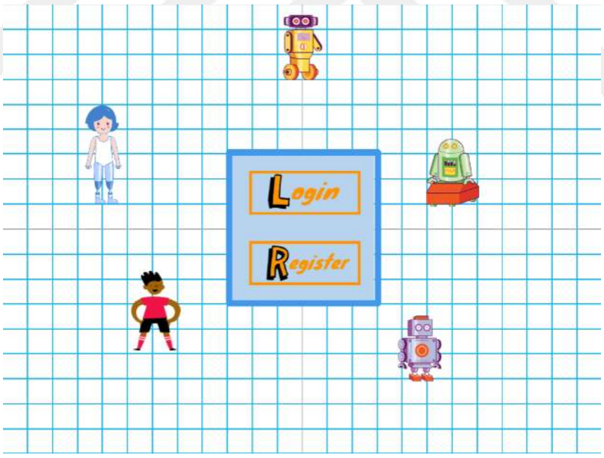

Figure 27: LRwtG Login page screen

Users who have successfully completed User Registration and logged into the system will be taken to the main game screen. This screen is shown in Figure 28.

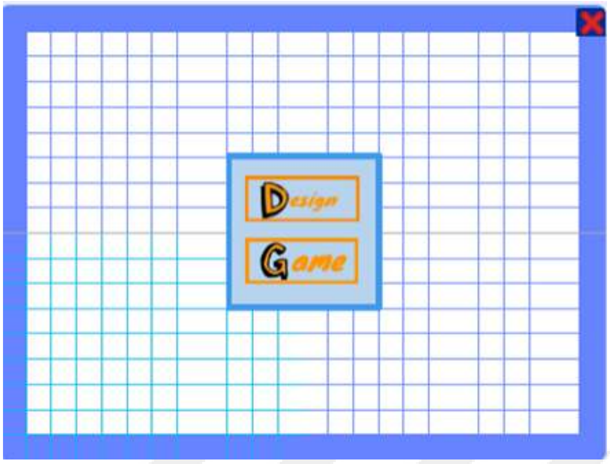

Figure 28: LRwtG Main page screen

Users can select one of the game or design options presented on this screen. Selecting one of the design or game buttons will take the user to a different page depending on their choice. While users who select the Design option will be greeted with the screen shown in Figure 29.

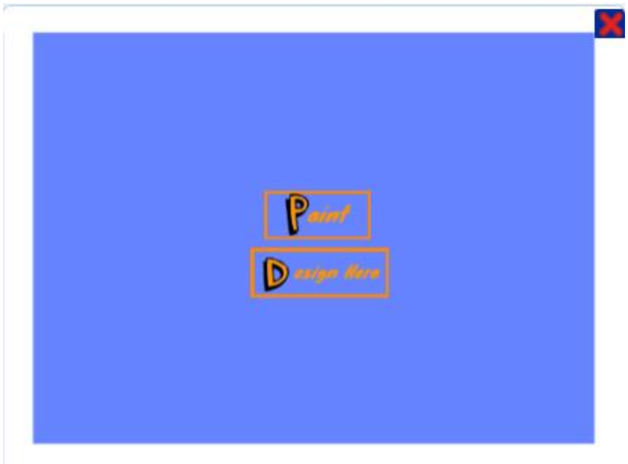

Figure 29: LRwtG Design page screen

Users who select the Game option that is on main page will be greeted with buttons containing 3 different games as seen in Figure 30. As the game progresses, users are presented with three different game screens.

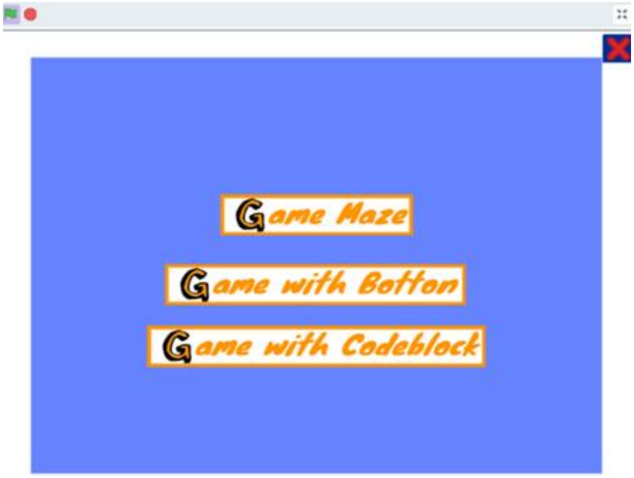

Figure 30: LRwtG Game page screen

The first game Game Maze button is clicked and the game screen displayed on the page Figure 31 is shown to the users. This game screen is similar to the normal maze games that are known by everyone today. In this game, it is requested to use the direction keys on the keyboard to move the puppet (character). As the direction buttons are pressed, the puppet starts to move and when it touches the orange wall, it gives an error message as in Figure 32. When the puppet reaches the star, the game ends and points are earned.

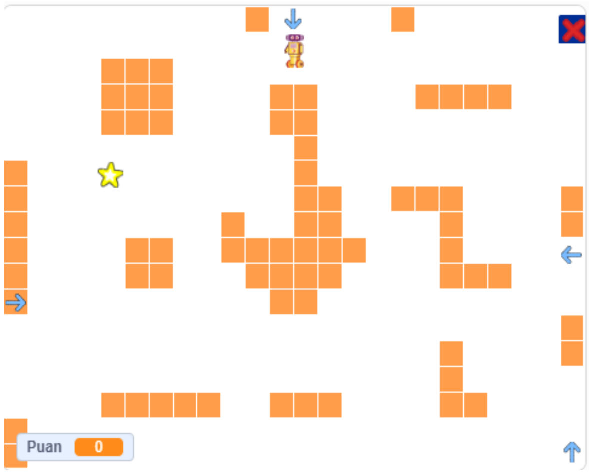

Figure 31: LRwtG Game Maze page screen (First Game)

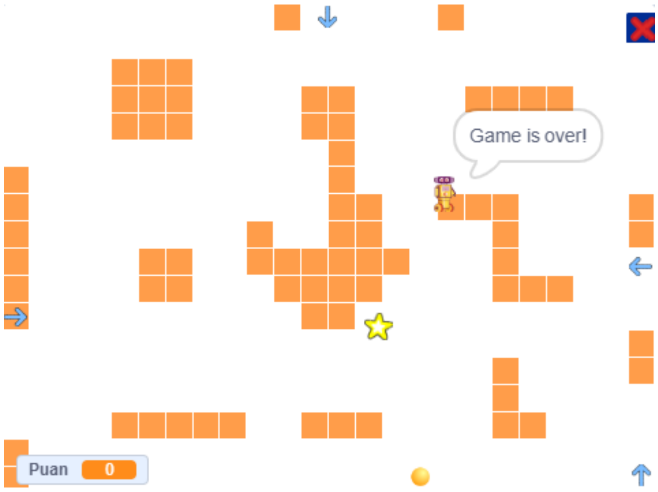

Figure 32: LRwtG Game Maze page error message screen

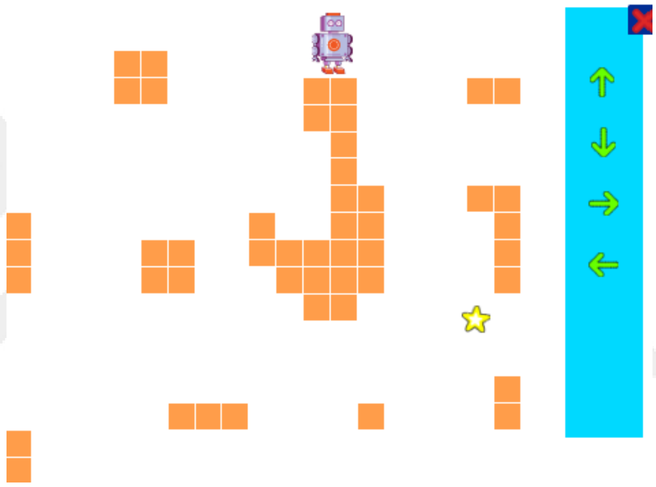

Figure 33: LRwtG Game with Button page screen (Second Game)

The second game screen is accessed by clicking the Game with botton button. The screenshot shown in Figure 33 is the screenshot of the second game. The puppet (character) here starts to move by pressing the direction buttons on the screen. When the puppet reaches the star, it collects points and the game ends. When it touches the orange walls, an error message is given as shown in figure 32 and the game ends.

The third game screen has a more complex structure with game levels than the first and second game screens. Therefore, it is considered as the main game of this project. With this game, it is aimed to teach Scratch and coding logic to the users. While the first two games have simple game play, this game requires a little more thinking and problemsolving skills. When the Game with codeblock button (in Figure 30) is pressed, firstly a screen explaining the purpose of the game as shown in Figure 34 welcomes the users.

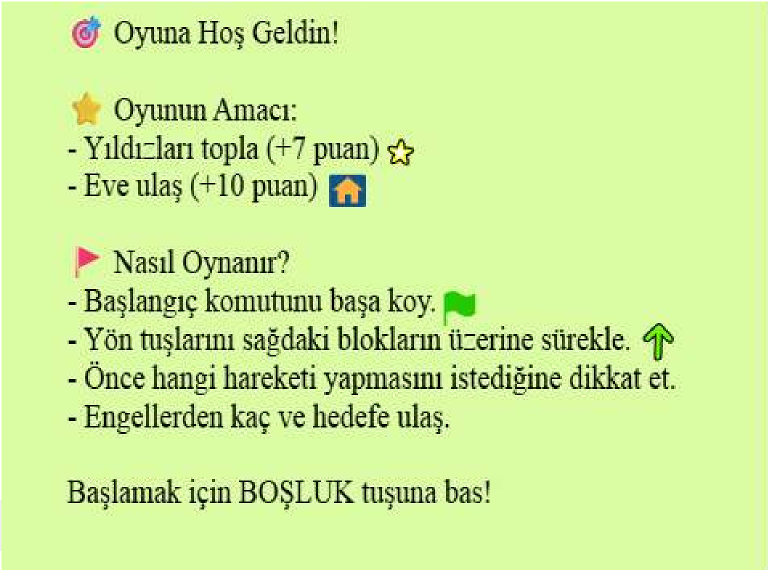

Figure 34: LRwtG Game with Codeblock welcome page screen

Figure 34 welcomes the user to the game and explains what the goal of the game is. Hints such as what is expected of the user and how to move around in the game are given, while information about what the user can do to collect points is shared.

In this screen (Figure 34), users are told that if they collect the star, they will get 7 points and if they reach the house icon, they will get 10 points and complete the game level.

The third game screen is the most educationally rich component of the LRwtG application. This game type is designed to provide users with a real coding experience. The game has the following features:

- Level-based progression system: Starting from simple tasks, users face increasingly more complex coding challenges.
- Drag and drop interface: Mirroring Scratch's block-based coding logic, this interface allows users to program by dragging and dropping blocks of code.
- Visual feedback: Users can instantly see the effect of their code on the robot, which encourages learning through trial and error.

- Algorithmic thinking skills: The game develops algorithmic thinking skills by requiring users to create step-by-step instructions to solve a problem.
- Loop and condition structures: At advanced levels, users are introduced to the basic programming concepts of loops and conditional statements.
- Goal-orientated tasks: At each level, the user needs to get the robot to a specific goal, which develops problem-solving skills.

This type of game allows users to learn robotic coding concepts in a fun and interactive environment. Users learn how to create the correct sequence of commands to get the robot to the goal, while also grasping the basic logic of programming. As the level progresses, users encounter more complex tasks and apply more advanced coding skills.

After the game objective screen (Figure 34) welcomes the users, the game is started by pressing the space bar on the keyboard and the level map shown in Figure 35 welcomes the users. All levels in this game are shown to the users on this map. The game levels start from the basic level and are completed with games explaining for loops and if-else loops. Thanks to this map, users can either play starting from the beginning or play the game levels in a mixed way.

Figure 35: LRwtG Game with Codeblock level map

When the 1st level button is clicked on the level map screen, the first level game screen, which is the first stage of the game, is shown to the users (see Figure 36). When the game screen is opened, the character tells the users in speech bubbles the clues of what to do in the game. On this screen, the user is asked to fulfil the conditions that the puppet (character) must fulfil in order to move. The green direction symbols at the bottom of the game screen and the green flag symbol are expected to be placed on the yellow blocks on the right side.

Figure 37 The conditions required for the first level game have been fulfilled correctly and the puppet is expected to perform the given commands by pressing the start button. The puppet shows the speech bubble indicating that it has started to move. If there was no green flag on the first yellow block, the puppet would not start moving and would give a warning. Because the green flag is the first command that must be given for the code to start running in scracth coding language. Therefore, it is important to start with the green flag first.

In the first level game, as shown in Figure 36, 3 yellow blocks are given on the side for the puppet to move. When the green flag icon, right direction arrow sign and right direction arrow sign are dragged into these 3 yellow blocks respectively, the puppet is ready to move as shown in figure 37.

Figure 36: LRwtG Game with Codeblock level one

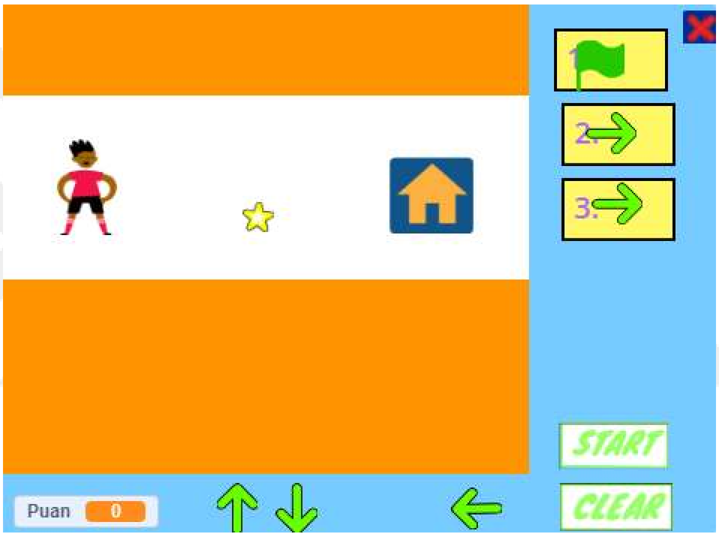

Figure 37: LRwtG Game with Codeblock level one – ready to move

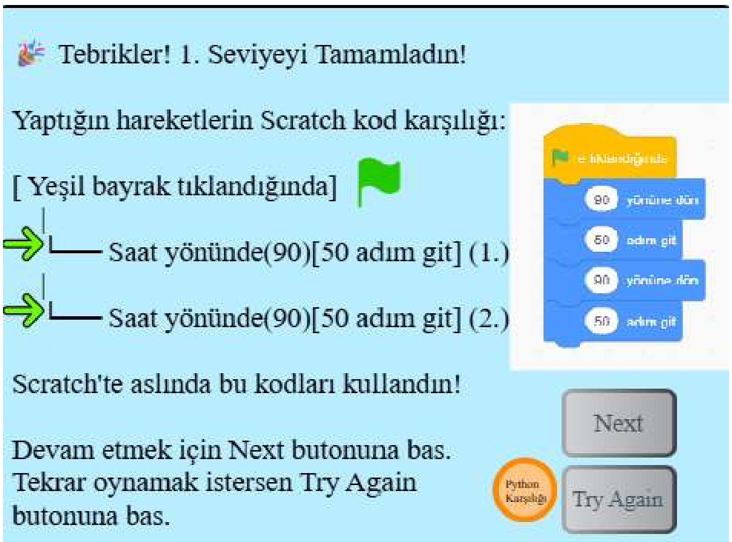

Figure 38: LRwtG Game with Codeblock - completed the first level

Finally, the start button must be clicked for the puppet to start moving. After the button is clicked, the puppet reaches the house icon by making the movements in the 2nd yellow block and 3rd yellow block respectively by releasing the speech bubble. Before reaching the house icon, if there is a star on its way, it collects the star and provides an increase in points. After reaching the home icon, the information screen in Figure 38 is displayed on the user's screen. On this screen, the coding language equivalent of the operations done in scratch language is shown to the users. After this stage, the user can play the game again or move to the next level. The python equivalent of the scratch code displayed on this page can be viewed by clicking the orange button. When this button is clicked, users are greeted with a screen in Figure 39.

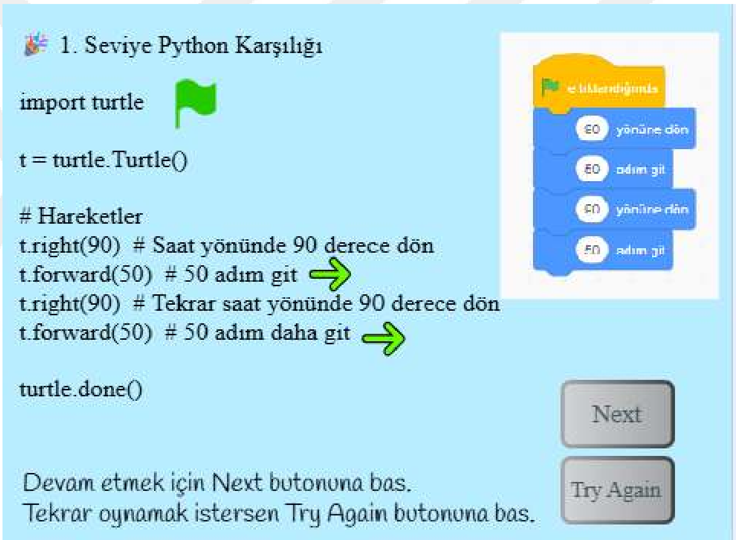

Figure 39: LRwtG Game with Codeblock - completed the first level - code equivalent in python language

When all levels are completed, the user is shown the information screens as shown in Figure 38 and Figure 39, which explain the code equivalents of the operations performed at that level.

Figure 40 shows a screenshot of the second level game. Here, the user received an error message because the user clicked the start button in the wrong order. The error here shows the error given when the user does not start with the green flag. In this case, the user clicks the clear button to return all icons to their original positions and restarts the game.

Figure 40: LRwtG Game with Codeblock level two

When the user has the correct order, the screenshot of the puppet's home icon is shown in Figure 41. Immediately after this screen, the user is shown the 2nd level completed notification screen, just like after level one (Figure 42).

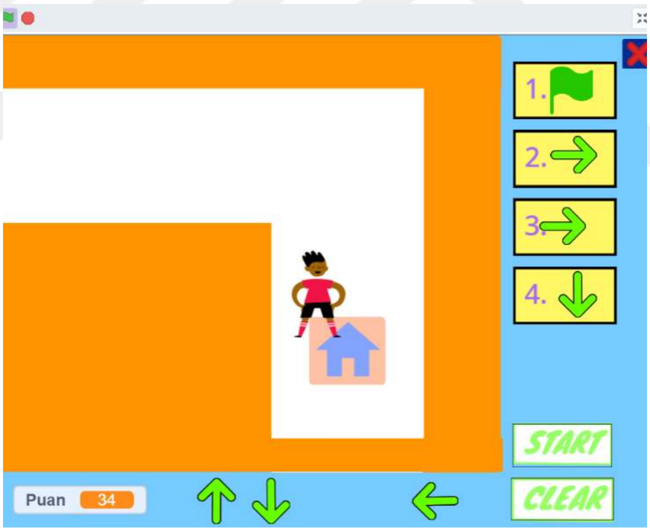

Figure 41: LRwtG Game with Codeblock level two- won screen

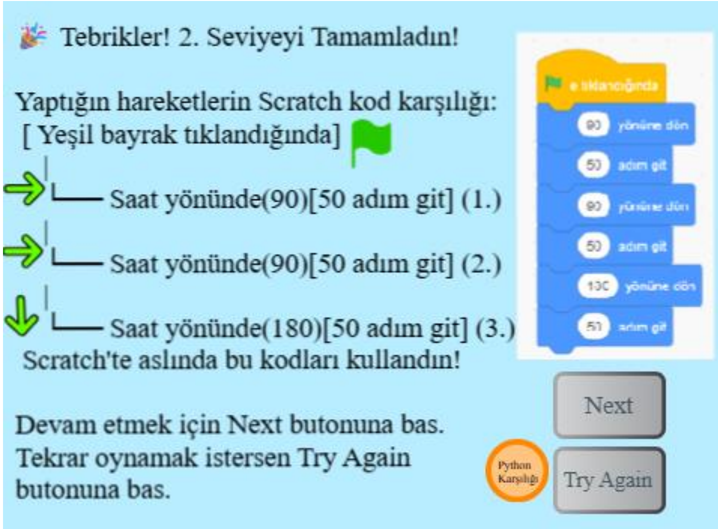

Figure 42: LRwtG Game with Codeblock - completed the second level

After the second level completion screen (Figure 42), the user can view the python language equivalent of the scratch codes in Figure 43 or access the third level (Figure 44) by clicking the next button.

Figure 43: LRwtG Game with Codeblock - completed the second level - code equivalent in python language

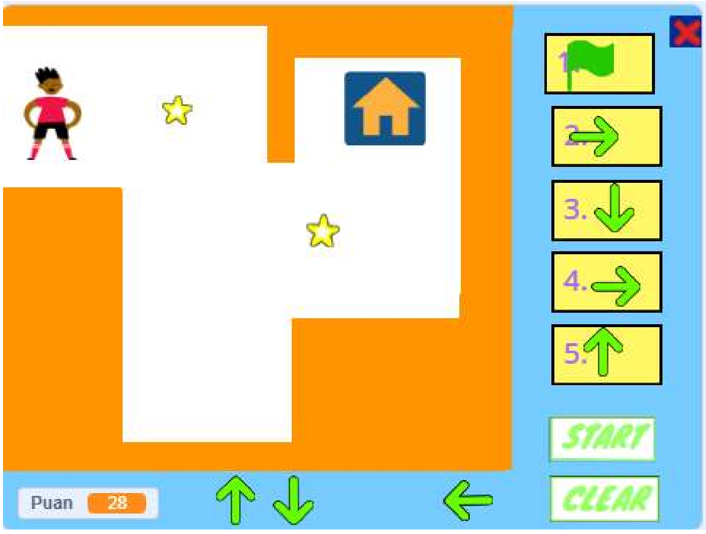

Figure 44: LRwtG Game with Codeblock level three

Figure 44 shows the third level of the game. In this screen, the sequence requested from the user has been realised but the start button has not been clicked yet. In this level, there are two stars and the puppet is waiting to reach home by collecting the stars. The green flag is placed first. In the second row, the right direction arrow icon, in the third row the down direction arrow icon, in the fourth row the right direction arrow icon and finally in the fifth row the up-direction arrow icon is placed to complete the game. After the start button is pressed, the puppet performs these steps one by one and reaches the house icon. After reaching the house icon, the information message is broadcast as in the first and second levels as shown in Figure 45.

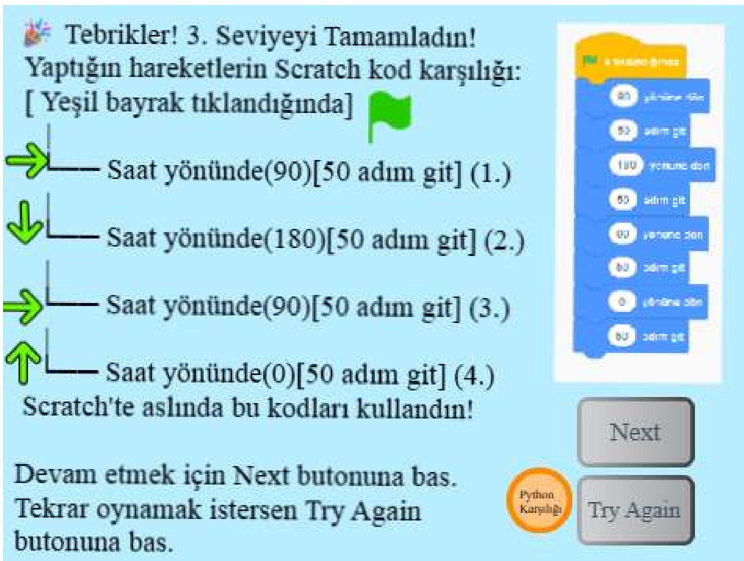

Figure 45: LRwtG Game with Codeblock completed the level three

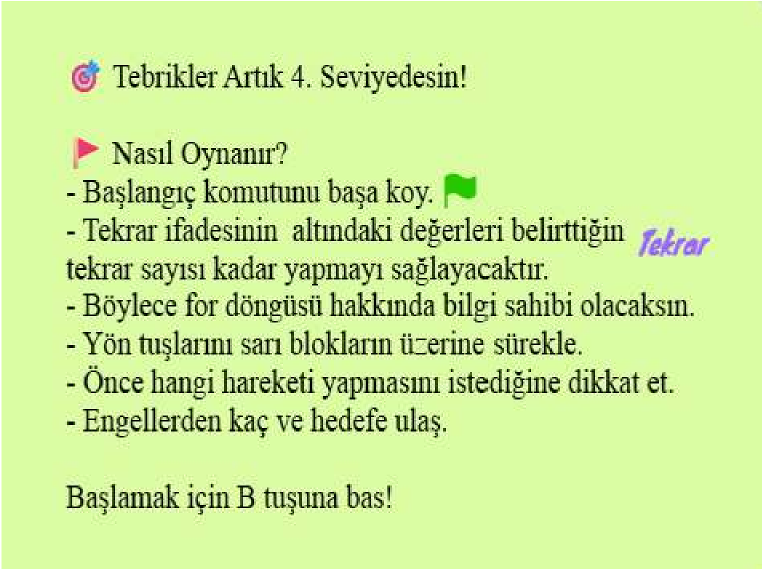

Figure 46: LRwtG Game with Codeblock - level four game objective sheet

After the completion of the 3rd level of the game (Figure 45), it is time to play the fourth level of the game. In this game, the level in which the for loop is aimed to be taught is started. Firstly, as shown in Figure 46, the screen that will provide information about the purpose of the game and how it is played welcomes the users. In this game level, there is a block that writes again. With the direction icons added under the block that says repeat, the repetitive movement we want the puppet to do is determined.

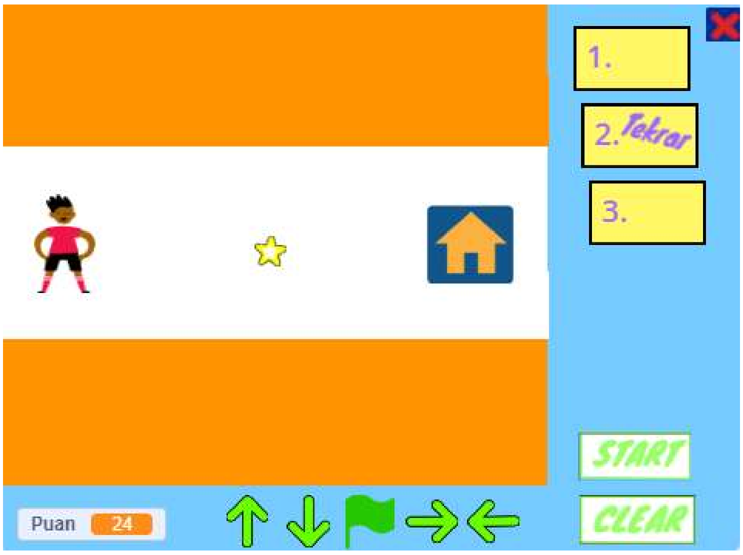

Figure 47: LRwtG Game with Codeblock level four

Figure 47 shows the 4th level game screen. In Figure 48, the green flag and the direction icon are placed on the yellow block and the user is asked how many times he/she wants the character to repeat. Here, the user is expected to enter a number from the keyboard in the white box below. In the system, which is restricted to enter data between 2 and 4 on the keyboard, the puppet moves in the right direction 3 times in line with the information entered by pressing the 3 key on the keyboard and reaches the home icon and the information message in Figure 49 welcomes the users.

Figure 48: LRwtG Game with Codeblock level four- the number of repetitions is asked

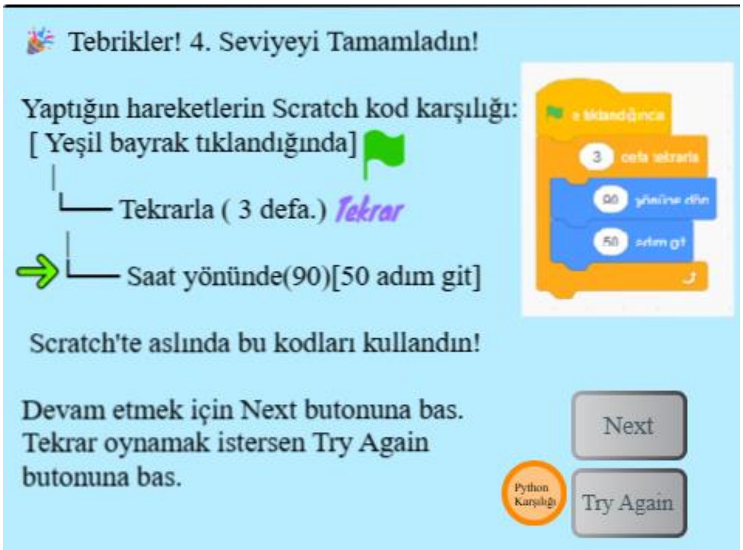

Figure 49: LRwtG Game with Codeblock completed the level four

In Figure 50, the equivalent of the operations performed in python language is presented to the users. If a different value is entered from the keyboard, an error message is given and the user starts the game again.

Figure 50: LRwtG Game with Codeblock completed the level four -code equivalent in python language

When we move to the fifth level, the middle level for loop game, which will challenge the users a little more, welcomes the game. At this level, a screen like the one in Figure 51 is displayed. This time the game screen resembles a ladder. There are two blocks under the repeat block and it shows that there are two movements that we want to repeat.

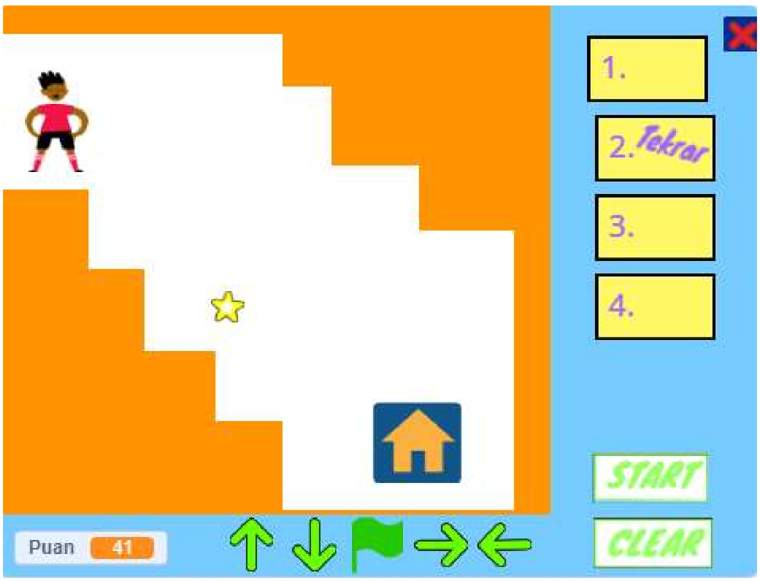

Figure 51: LRwtG Game with Codeblock fifth level

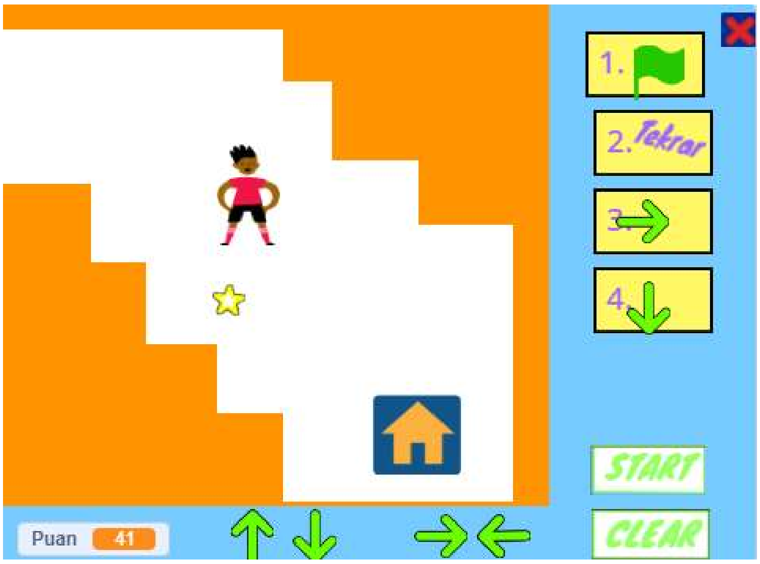

Figure 52: LRwtG Game with Codeblock fifth level – moving

In Figure 52, the desired movement blocks were listed and the character started to move. At this level, as in the previous level, the user was asked how many times he/she wanted the puppet to move and the puppet started to move. After the number 4 entered from the keyboard, the movement in Figure 52 is displayed. When the puppet reaches the home icon, the information screen in Figure 53 welcomes the users.

Figure 53: LRwtG Game with Codeblock completed the level five

The sixth level is the third game for the for loop. In this game, as shown in Figure 54, there are two repeat expressions and the user is asked for the number of repetitions separately.

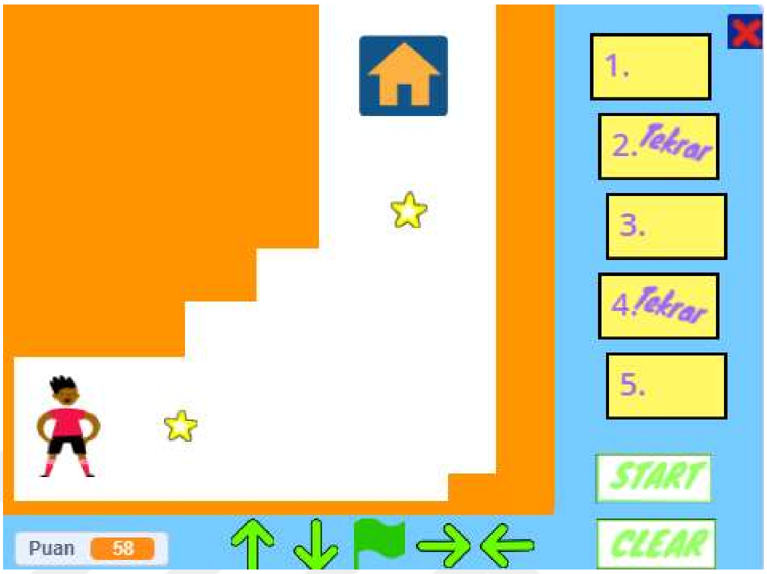

Figure 54: LRwtG Game with Codeblock level six

Figure 55: LRwtG Game with Codeblock level six – second the number of repetitions is asked

In Figure 55, after completing the first number of repetitions, the second number of repetitions is asked to the user. At this level, there is a data entry restriction between 2 and 4. If the user enters a different value, the level is repeated. When the dummy reaches the home icon, the information message in Figure 56 welcomes the user.

Figure 56: LRwtG Game with Codeblock completed the level six

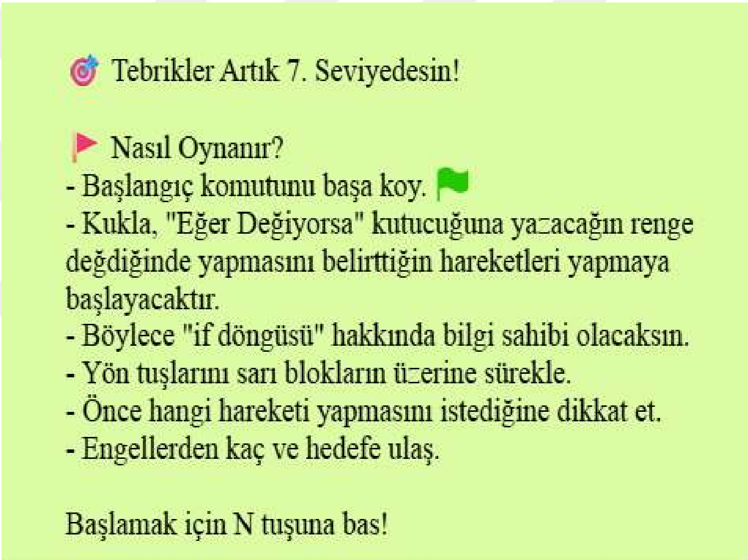

Figure 57: LRwtG Game with Codeblock- level seven game objective sheet

In the seventh level, it is desired to inform the users about the if loop. When this level is passed, an information screen as shown in Figure 57 is displayed. Here, information about how to play the game is given. This level is slightly different from the other levels. Because here, if the puppet touches the star, we will be listing the actions we want the puppet to do as a result of the statement. Figure 58 shows the game screen that welcomes us at the seventh level.

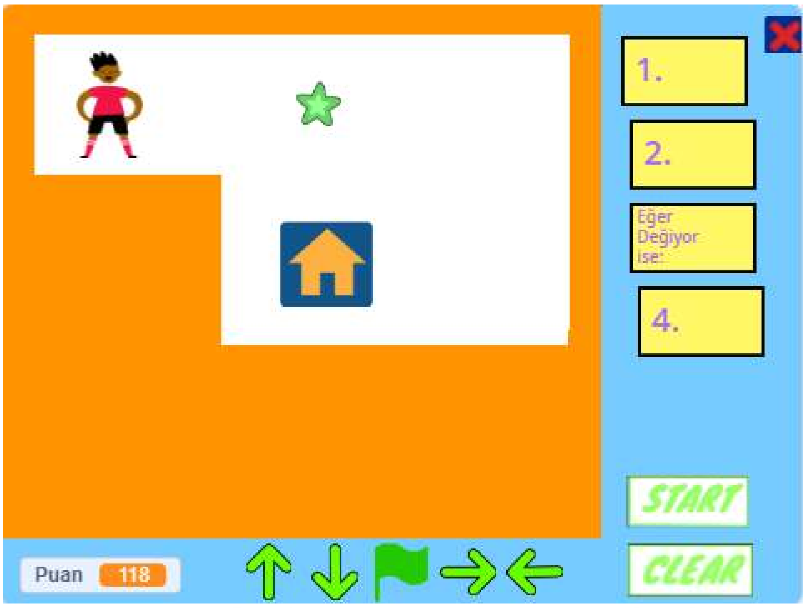

Figure 58: LRwtG Game with Codeblock level seven

Figure 59: LRwtG Game with Codeblock level seven – asked star colour

As shown in Figures 58 and 59, block 3 contains the expression 'if it touches'. When the steps like in Figure 59 are completed and the start button is clicked, the question 'which colour does the puppet move if it touches?' is asked. When the expression 'yesil' or 'yeşil' is written in the white box below in Figure 59, when the puppet touches the green star, it will make the movement of the direction icon in the 4th block.

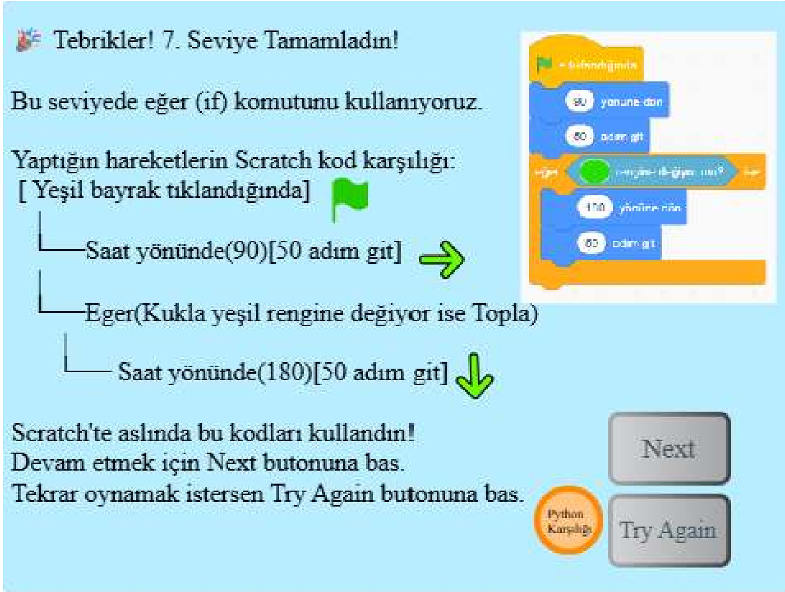

Figure 60: LRwtG Game with Codeblock completed the level seven

If the puppet is worth the green star, after you go down, you both collect the star and earn points and reach the game level is completed. After the level is completed, the information display in figure 60 meets the users. As in all levels, this level is shown to users in exchange for the python language. For this, it is enough to click the orange button.

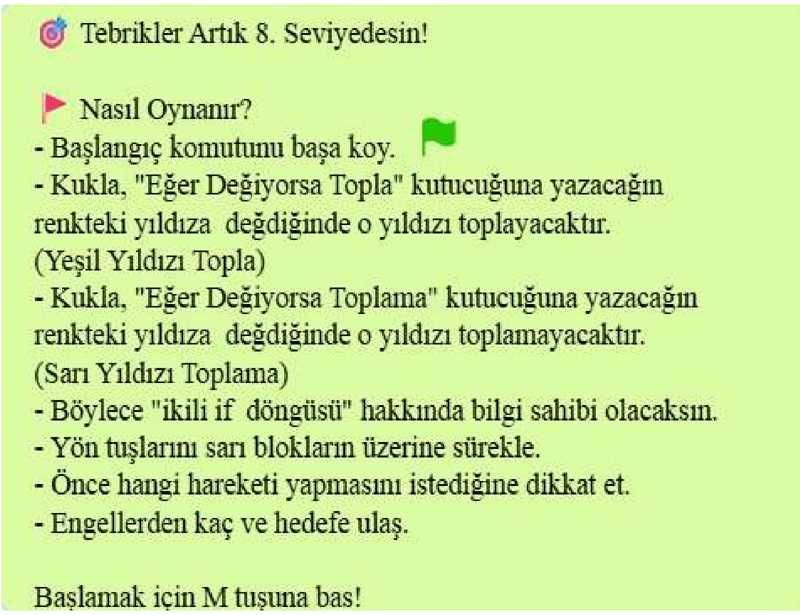

Figure 61: LRwtG Game with Codeblock level eight game objective sheet

In the eighth level, there is a game about the if cycle again at this level, and in the game at this level, the focus is not on changing the direction as in the seventh level, but on collecting the star or collecting. Figure61 also provides information about this game. Figure 62 shows the game screen at the eighth level.

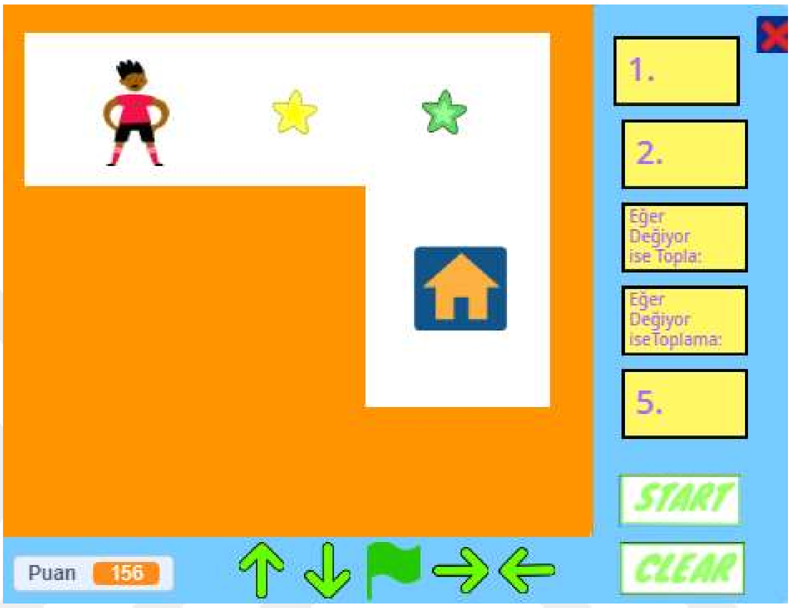

Figure 62: LRwtG Game with Codeblock level eight

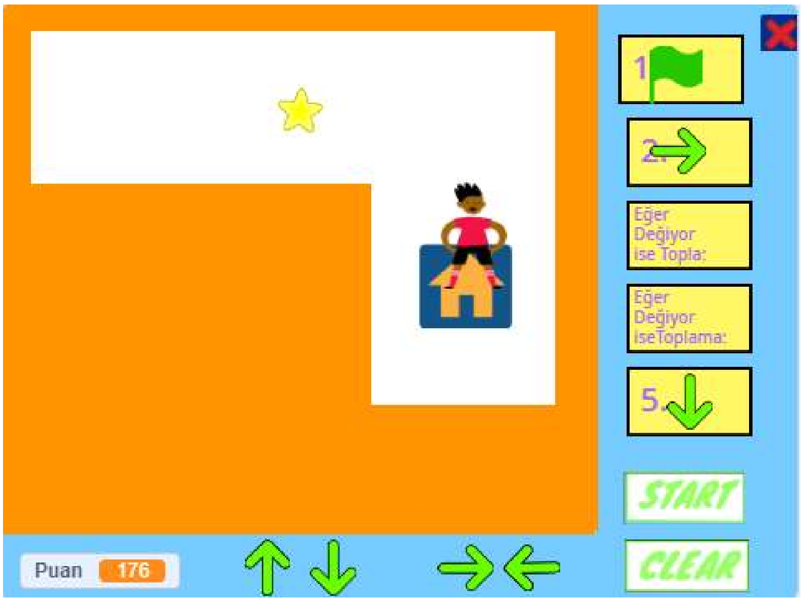

Figure 63: LRwtG Game with Codeblock level eight won screen

The third yellow block contains the phrase "collect if it is worth it". Users are expected to enter the phrase "energetic" here. 4. the yellow block contains the phrase "collection if it is worth it". Users are expected to enter a "yellow" statement here. When they enter an incorrect value, an error is written. When these operations are performed, they are seen as in figure63. After figure 63, it meets the users figure 64. The provisions of the transactions are shown in the scratch language.

Figure 64: LRwtG Game with Codeblock completed the level eight

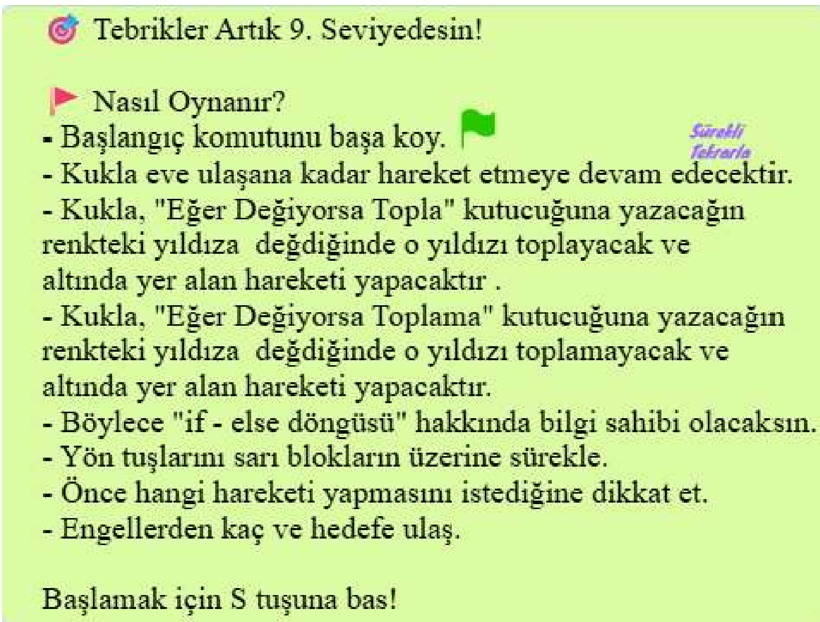

Figure 65: LRwtG Game with Codeblock level nine game objective sheet

The game purpose page, which tells how to play for the touch level, which is the last level of the game, and what the purpose in the game is, is shown to users as in figure 65. At this level it is aimed to teach the if-else cycle. After pressing the "S" key from the keyboard, its users are greeted by the figure 66 screen. Here, the expression "continuously repeat" on the yellow blocks indicates that the transactions will continue until the puppet reaches the house icon.

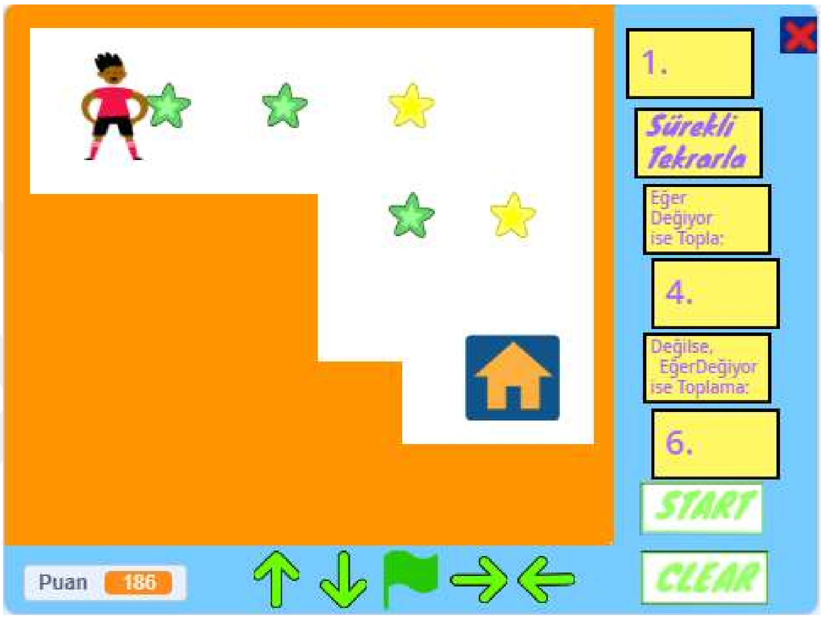

Figure 66: LRwtG Game with Codeblock level nine

Figure 67: LRwtG Game with Codeblock level nine – asked star colour

If the expression "collect if it is worth" on the yellow blocks is "green", the puppet will collect the star and make the movement in the block below it. 5. after the "yellow" color to be written in the "if not the summation" statement in the yellow block, the puppet will not collect when it touches the yellow stars, but will make the movement at the bottom. In Figure 67, the 3rd block "move what color it's worth?" a screenshot is shown when asking the question.

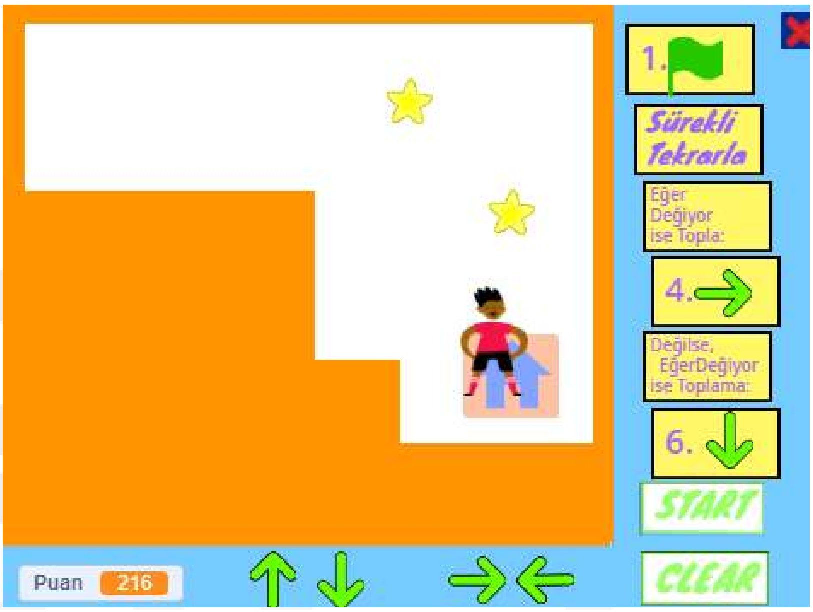

Figure 68: LRwtG Game with Codeblock level nine won screen

After the "masculine" and "yellow" expressions are written from the keyboard, respectively, the puppet reaching the target is displayed as in figure 68. After figure 68, the screen in figure 69 is shown to users autonomously. On this screen, the value of the transactions in the scratch language is shown to the users.

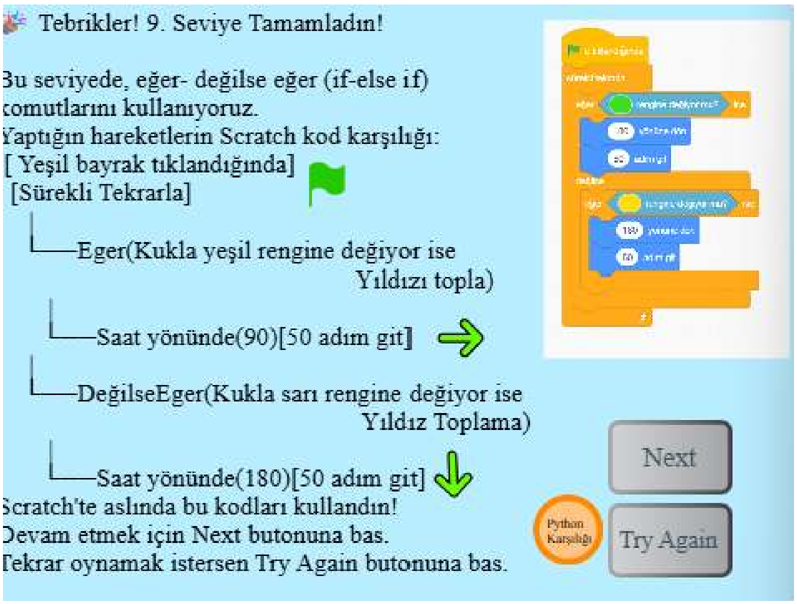

Figure 69: LRwtG Game with Codeblock completed the level nine

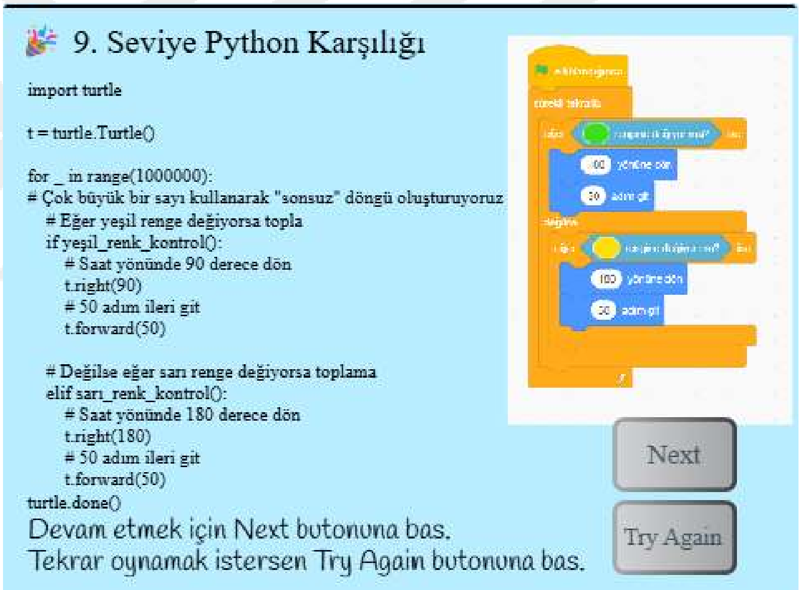

Figure 70: LRwtG Game with Codeblock completed the level nine-code equivalent in python language

Figure 70 shows the equivalent of the game in the python language in scratch. Finally, when this game level is completed and click on the next button, users have completed all levels greeting the written figure 71 meets the screen.

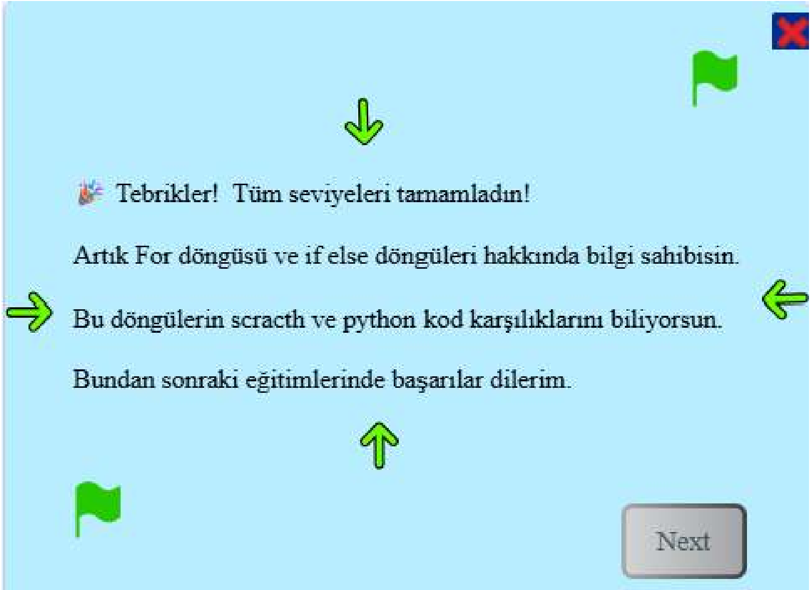

Figure 71: LRwtG Game with Codeblock - All levels are complete congratulations screen

In addition to the games in the application, users are also given the opportunity to design their own characters. To access this section, users just need to click on the DESIGN button on the main page. The character created here can then be used in the third game (Code block game) screen to continue the game and the game can be played repeatedly with new character designs. Users who reach the character design page can change the head, body and foot parts of the puppet by using the buttons at the bottom of the page in Figure 72.

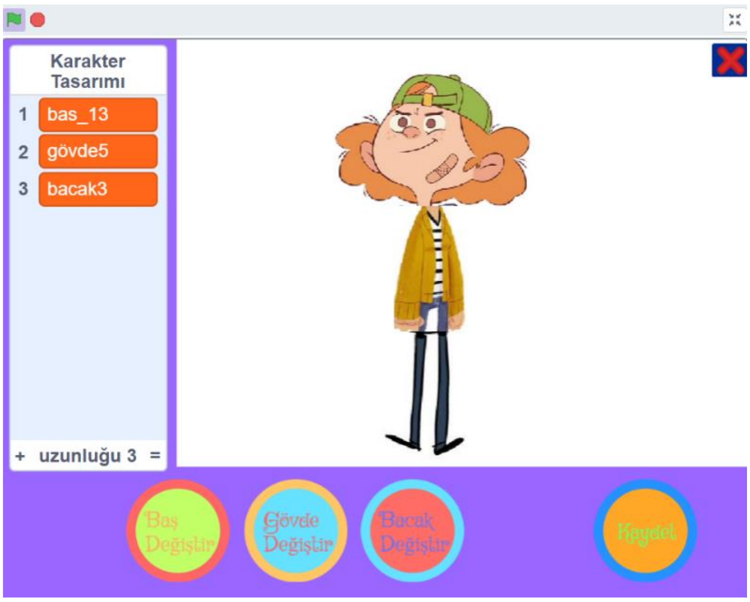

Figure 72: Character design page screen

After making as many changes as desired, the user who clicks the SAVE button saves the puppet to the character design list. The puppet design information saved in the character design list will be displayed when returning to the third game screen. The user can play the game with the fixed character or create new designs and play the game levels with the character he/she designed.

SECTION THREE
RESEARCH FINDINGS AND DISCUSSION

The gamified educational platforms can be used to make the educational process more fun and efficient by differentiating it from traditional educational methods. The Scratch application has been implemented in numerous educational settings. The Scratch application in this thesis has been developed to present robotic coding knowledge in a more engaging and enjoyable manner. In addition, the ability to continuously develop these applications is a significant advantage.

Respectively, under the first following title, the comparisons of games related with this study and the comparisons of the examined theses studies are given. Second title gives the analysis about the LRwtG Game Assessment Survey.

3.1 Literature Review and Game Platform Analysis Findings

The tables presented herein facilitate a clearer understanding of technological diversity and methodological approaches to research in the field of robotic coding with educational games.

Table 2 gives us a comparison of the educational games platforms that are in current use. It indicates that the platforms employed in the context of programming education exhibit a propensity to adopt diversified teaching strategies, according to the ages of the target population. In the early stages of childhood, specifically from ages 4 to 6, it was observed that educational platforms such as Kodable, Code.org and Code Karts seek to instruct fundamental programming principles through the utilisation of visual and intuitive interfaces, which are underpinned by minimal reading abilities. At primary school level (ages 6-10), the Code Monkey and Tynker platforms are defined as applying hybrid teaching methodologies that provide a gradual transition from the paradigm of visual programming to the text-based coding approach. At the middle school level (10-14 years old), Scratch Labyrinth

Game, Blockly Games and CodeCombat platforms have been developed to maintain student motivation by presenting complex programming issues in gamified and interactive learning environments.

Table 2: Comparison of the Popular Educational Coding Platforms

Platform Name	**Target Age Group**	**Coding Language Learning**	**Programming Approach**	**User Experience Design**	**Key Features & Capabilities**	**Educational Focus & Learning Objectives**
Robocode	12+ years (Advanced)	Java, .NET	Text-based Programming (Java, .NET)	Competitive programming environment with virtual robot battles	Real-time strategy gameplay, artificial intelligence programming, tournament system, advanced debugging tools	Advanced programming concepts, AI algorithm development, object-oriented programming, competitive coding skills
Code Monkey	6-14 years	CoffeeScript (Python-like syntax)	Text-based with CoffeeScript-like Syntax	Game-based progressive learning with immediate visual feedback	Real code writing experience, banana collection challenges, step-by-step skill building, visual code execution	Transition from visual to text-based coding, syntax understanding, logical thinking, problem decomposition
Coding Game	10+ years	25+ languages (C, C++,	Text-based, puzzle/escape/ game-based, competitive	Multi-language support, code block	Puzzles, escape rooms, multiplayer, code samples,	Syntax mastery, algorithmic thinking,

		Python, etc.)		interface, sample code, competitive play	language selection, international competitions	competitive programming, multi-language exposure
Scratch Labyrinth Game	7-14 years	Block-based (Scratch)	Visual block coding, game creation	Drag-and-drop interface, maze/puppet creation, creative project building	Game/animation creation, background/character design, step-by-step guides	Creative thinking, basic programming logic, event-driven programming, project-based learning
Blockly Games	8-14 years	Block-based, JavaScript transition	Block-based with Text Transition	Progressive puzzle-based learning system	Gradual complexity increase, seamless transition from blocks to JavaScript, multiple game types (Maze, Bird, Turtle)	Programming concept mastery, logical problem-solving, preparation for text-based coding, algorithmic thinking
Tynker	5-14 years	Block-based, Python, JavaScript, etc	Hybrid: Block-based to Python	Comprehensive guided learning with multi-platform support	Structured courses, game creation tools, mobile app compatibility, teacher dashboard, progress tracking	Complete programming education pathway, STEM integration, game development skills, creative project building
Code Combat	10+ years	Python, JavaScript, Lua	Text-based (Python, JavaScript)	RPG-style adventure with coding	Character progression system,	Professional programmi

				mechanics integration	multiplayer competitive arenas, real syntax practice, fantasy adventure storyline	ng languages, algorithm implementation, syntax mastery, collaborative coding
Code.org	4-18 years	Block-based (young), JavaScript, HTML, CSS (older)	Age-appropriate: Block-based to Text	Structured curriculum with comprehensive age-level progression	Hour of Code global initiative, classroom integration tools, teacher resources, diverse activity types	Computer science fundamentals, digital citizenship, inclusive programming education, curriculum alignment
Crunchzilla (Code Monster)	8-14 years	JavaScript	Text-based, interactive tutorial, immediate feedback	Simple interface, live code results, lesson progression, visual output	Animation, fractals, variables, loops, conditions, functions, instant visual feedback	Basic programming concepts, exploration, experimentation, immediate cause-effect understanding
Code Karts	4-8 years	Visual (directional blocks)	Block-based, logic sequencing, racing game	Colorful, car racing, single/multiplayer, increasing difficulty	Directional logic, single/two player, progressive levels, mobile/tablet support	Sequencing, logic, direction, early computational thinking, problem solving
Kodable	4-10 years	Block-based (visual)	Visual Block-based Programming	Colorful, engaging maze-based	Teacher curriculum integration, parent progress	Early programming logic, sequence

				challenge system	dashboard, offline activity support, character-driven learning	understanding, loop concepts, foundational computational thinking

In terms of programming approaches, analysis has facilitated a shift towards logical thinking processes by eliminating syntax errors through drag-and-drop interfaces of block-based visual programming platforms (like Scratch, Kodable, Blockly Games). This pedagogical approach has been attenuating the cognitive burden experienced by students. Conversely, text-based programming platforms (e.g. Robocode, CodeCombat) seek to equip students with the skills necessary to thrive in professional software development environments. These platforms focus on the instruction of industrial programming languages such as Java, Python and JavaScript. Moreover, it has been observed that platforms adopting a hybrid approach (like Tynker, Code Monkey, Blockly Games) provide a gradual transition from visual elements to text-based coding concepts. An analysis of the chronological distribution of the studies examined and the research trends revealed a significant increase in the field of game-based programming education in recent years.

In order to facilitate a comparison of the relevant literature in the field of robotic coding education by educational games, Table 3 is given below. About the related thesis studies, Table 3, provides the information like year, arget audience, platform, used technology, research method, key findings of the study and the contribution of the study to the field.

Table 3: Comparison of Thesis Studies on Scratch-Based Coding Education

Author & Year	**Study Title**	**Target Audience**	**Platform / Technology Used**	**Research Method**	**Key Findings**	**Contributions to Education**
Dizman, 2018	The Effects of Coding, Robotics, 3D Design and Game Design Education on the 11-14 Years of Students' ProblemSolving Skills and Level of Metacognitive Awareness	Middle School Students	LEGO Mindstorms EV3	8-week Experimental Design with 4C Approach	Robotics education improved mathematical thinking, collaborative work, innovation and problem-solving abilities	Supports 21st-century skills development through tangible programming experiences
Egin, 2019	An investigation of information technologies teachers' opinions on coding teaching	Information Technology Teachers (105 teachers)	Block-based and robotic coding tools	Mixed Method within Kodla(Ma)nisa Project	Teachers preferred block-based and robotic writing tools, used visual and game-based teaching methods	Guides effective coding instruction methods and tool selection for educators
Konyaoğlu, 2019	The Effect of Robotic Coding Education on the Problem-Solving	Secondary School Students	Edison educational robots, Makeblock mBot robots	Experimental Design	Five-week robotics training programme had substantial and	Demonstrates effectiveness of short-term robotic coding

	Skills of Secondary School Students and The Views of the Students About Robotic Coding Activities				favourable influence on problem-solving abilities	interventions
Eraytaç, 2019	The effect of block-based teaching method on academic achievement of secondary school students in robotic coding training	130 Secondary School Students	mBlock (Block-based) vs Arduino (Text-based)	Semi-Experimental Post-test Design	Block-based (mBlock) group demonstrated higher academic success than text-based (Arduino) group	Validates superiority of block-based approaches in robotics education
Canbeldek, 2019	Exploring the effects of productive children: Coding and robotic education Program in early childhood education	Children aged 5-6 years (80 children)	Coding and Robotic Education Program tools	Semi-Experimental with Control Group Design	Significant increases in cognitive development, language skills, and creativity scores	Establishes early childhood as critical period for coding and robotics introduction
Fidan, 2016	Effect of gamification in teaching programming with scratch on student engagement	Students	Scratch software with gamification elements	Qualitative Research Method	Gamification enhanced student engagement, motivation, and academic achieveme	Demonstrates effectiveness of gamification in programming education

					nt while making learning more enjoyable	
Totan, 2021	The effect of blockbased coding education on the students' attitudes about the secondary school students' computational learning skills and coding learning: Blocky sample	Middle School Students	Block-based coding platforms	Mixed Method Research	Substantial positive enhancement in self-efficacy and attitudes towards information-processing thinking	Shows positive impact on student confidence and learning attitudes

When we compare the related theses it is seen that the target audience distribution was determined that it covers a wide range of ages from preschool to university level. In parallel with the findings of Alper and Sanli (2024), meta-analysis findings, the studies focused mainly on secondary school students.

Analyses of technological platform preferences have revealed that Scratch is the most widely used tool. This finding supports the platform selection of the Scratch-based maze game developed in the current study. Scratch's block-based visual programming approach offers an ideal learning environment, especially for children ages 6-12. Eraytac's (2019) comparative study empirically proved that block-based coding tools are superior compared to text-based coding tools for students who are just starting out in programming.

Studies by Konyaoglu (2019), Dizman (2018) and Kaya et al. (2020) in terms of problem-solving skills have shown that robotic coding education makes statistically significant contributions to students' problem-solving capacities. These findings reinforce the theoretical underpinnings of the problem-solving-driven design of the maze game being developed.

Totan (2021) and Kaya et al. (2020) developed the core computing components of block-based coding education such as decomposition, pattern recognition, abstraction, and algorithmic thinking.

In terms of creativity development, the work of Haymana and Ozalp (2020) and Canbeldek (2019) empirically supported that robotic coding training positively affected students' capacity to think creatively.

In the dimension of motivation and student engagement, according to Fidan's (2016) qualitative findings, the integration of gamification elements significantly increased student engagement and intrinsic motivation. This finding supports the pedagogical value of the gamified structure of the Scratch-based maze game.

3.2 LRwtG Game Assessment Survey Analysis

According to the survey results which was applied to 12 participants LRwtG (Learning Robotics with the Game) game provides positive results. The results according to the survery questions are given below.

1. The contribution level of the game's potential to engage the students was found to be 4.08 on average out of 5, with 58.33% of participants rating it 4 points.
2. The contribution level of the game's educational aspect was found to be 4.25 on average out of 5, with 41.67% of participants rating it 5 points and another 41.67% rating it 4 points.
3. Participants rate the educational content of the game as 4.17 on average out of 5, with 41.67% of participants giving the highest score of 5.
4. The contribution level of the game to students' problem-solving skills was found to be 4.42 on average out of 5, with 58.33% of participants rating it the maximum 5 points.
5. The contribution level of the game to students' algorithmic thinking skills was found to be 4.42 on average out of 5, with 50% of participants giving it the highest score.
6. The contribution level of the game to students' coding skills was found to be 4.00 on average out of 5, with 50% of participants rating it 4 points.

7. Participants rated the game as being at a level that students can easily understand as an average score of 4.08 out of 5, with 41.67% giving it 4 points.
8. Participants rated the level of the fun of the game as 4.08 on average out of 5, with 58.33% of participants rating it 4 points.
9. Participants rated the level of the game's ease of play as 4.42 on average out of 5, with 50% of participants giving the maximum score of 5.
10. Participants rated the level of user-friendliness of the game's interface as 4.00 on average out of 5, with 66.67% of participants rating it 4 points.
11. Participants rated the level of game's technical features as 4.00 on average out of 5, with 66.67% of participants giving it 4 points.
12. Participants rated the level of the game's visual design as 4.08 on average out of 5, with 41.67% of participants rating it the highest score of 5.
13. Participants rated the level of the game overall as 4.25 on average out of 5, with 41.67% giving it 5 points and another 41.67% giving it 4 points.
14. Participants' recommendation rate for the game was found to be 4.25 on average out of 5, with 50% of participants giving the highest recommendation score of 5.

From these findings it is seen that the highest-scoring criteria were problem-solving skills (4.42/5.00), algorithmic thinking skills (4.42/5.00) and ease of play (4.42/5.00). These values can be evaluated as quite high. These findings show that the game's basic design goals have been successfully met. The fact that problem-solving skills score the highest confirms the effectiveness of the maze-based game mechanism. A high assessment of algorithmic thinking skills reflects the success of the block-based coding approach.

In contrast, the criteria with the lowest scores were determined as technical specifications (4.00/5.00) and potential for attracting interest (4.00/5.00). These results show that the game has the potential to improve the technical infrastructure and visual design elements.

When evaluated in terms of the level of recommendation, 50% of participants recommend the game at the highest level (5 points), which suggests that the educational value of the game is highly accepted. Of the remaining participants, 33.3% (4 participants) gave a recommendation score of 4 points, indicating strong approval but with some room for improvement. Additionally, 16.7% (2 participants) rated the game at a moderate recommendation level of 3 points, suggesting they found the game satisfactory but with notable areas for enhancement.

When the results of the evaluation on a category basis are analyzed, it is seen that the skill development category has the highest average with a score of 4.28/5.00. This category encompasses three critical dimensions: problem-solving skills (4.42/5.00), algorithmic thinking skills (4.42/5.00), and coding skills (4.00/5.00). The survey data reveals that 58.33% of participants rated the game's contribution to problem-solving skills at the maximum 5 points, while 50% gave the highest score for algorithmic thinking skills development. For coding skills, 50% of respondents assigned a score of 4, indicating strong but not maximum satisfaction. These findings collectively demonstrate that the strongest aspect of the game is its potential for children to develop computational thinking competencies, particularly in problem-solving and algorithmic thinking. This aligns with established research in the field (Kaya et al., 2020; Akgün, 2020; Erdel, 2024) that demonstrates how robotic coding training enhances computational thinking skills through structured learning experiences that require logical reasoning, pattern recognition, and sequential thinking.

The educational value category ranks second with a score of 4.21/5.00, comprising three essential educational components: the game's educational aspect (4.25/5.00), educational content (4.17/5.00), and the game's overall evaluation (4.25/5.00). Survey results indicate strong participant confidence in the game's educational merit, with 41.67% of respondents rating both the educational aspect and overall evaluation at the maximum 5 points, while another 41.67% assigned 4 points to each. Similarly, for educational content, 41.67% of participants gave the highest score of 5, with 33.33% rating it 4 points. This consistent distribution of high scores across all educational dimensions demonstrates that the game effectively balances educational objectives with engaging gameplay. These findings support the pedagogical effectiveness of Scratch-based educational practices (Strawhacker et al., 2018; Yukselturk & Altıok, 2016; Fidan, 2016), which emphasize visual programming environments as accessible entry points for developing computational thinking while maintaining educational rigor and content relevance.

The general category reveals that the attracting potential with a 4.19/5.00 point average is high in the overall level of evaluation and advice. The user experience category shows that it is well evaluated in terms of comprehensibility, entertainment level, ease of play and user-friendly interface with 4.15/5.00 points. The technical and visual category is rated slightly lower than the other categories with a score of 4.04/5.00 and points to areas that need to be developed. Figure 73 and figure 74 contain evaluation tables of data from survey analyses.

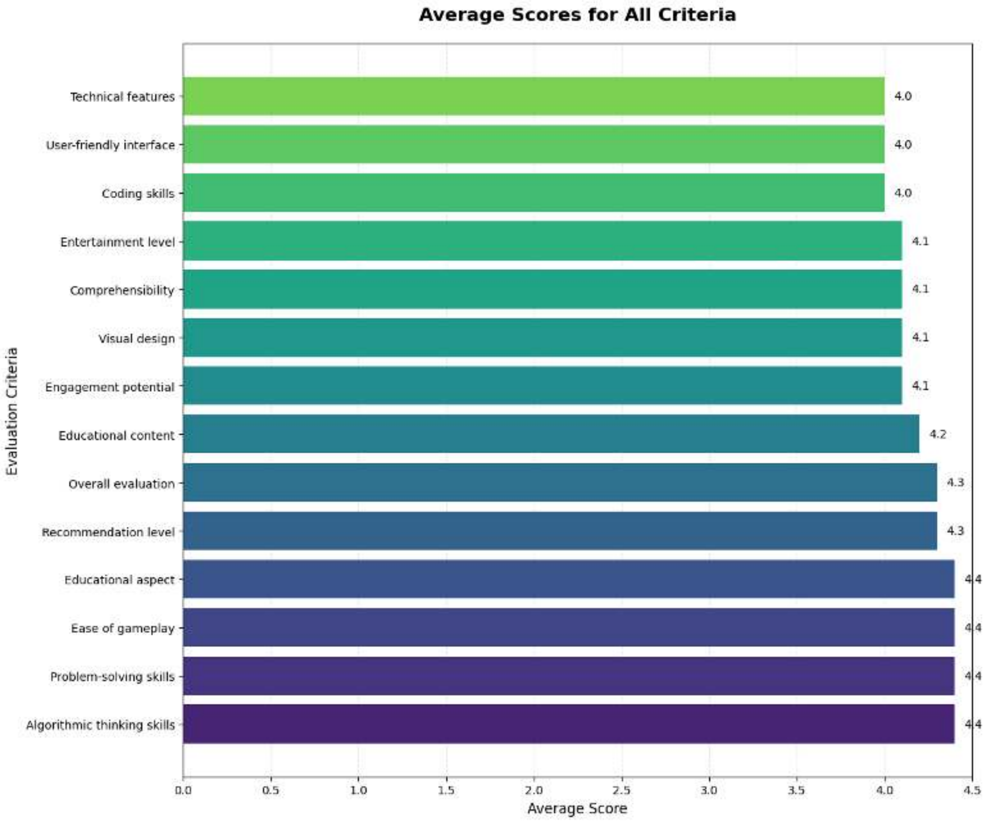

Figure 73: Criteria graph of average values of survey data

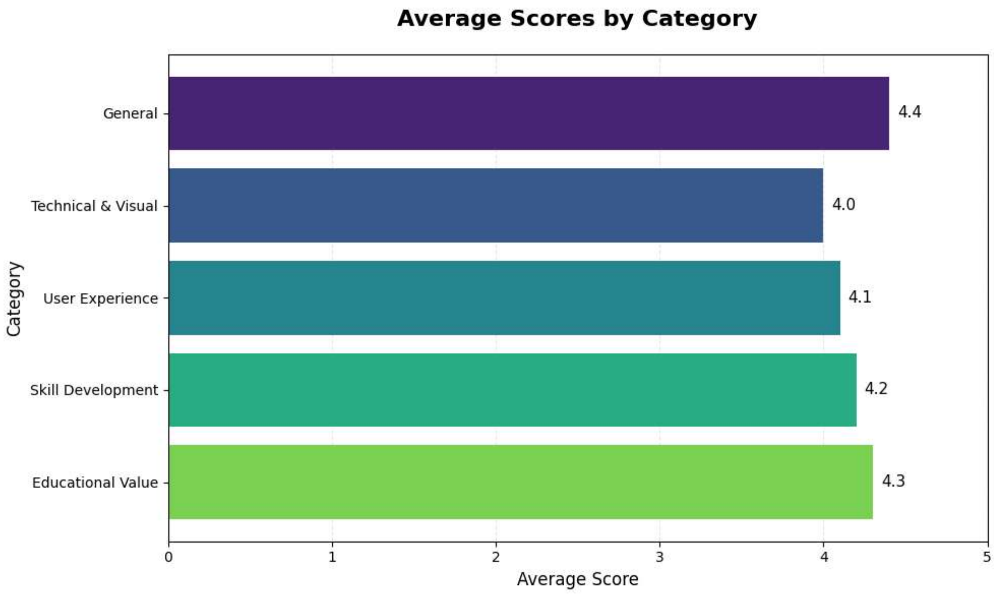

Figure 74: General valuation and categorization graph

The overall average of all the criteria is calculated as 4.18/5.00, which shows that the game largely meets the goal of teaching robotic coding for children aged 6-12 years who are the target audience. This conclusion is consistent with the literature findings that support the effectiveness of Scratch-based educational games.

The suggestions obtained from the open-ended comments of the participants clearly reveal the areas in which the game should be developed. these can be summarized as below:

- Within the scope of level system development, it was proposed to add different difficulty levels and to ensure a gradual increase in level difficulty.
- In terms of programming features, it is stated that the use of variables should be added, the inclusion of loop structures and the development of the block-based programming interface.
- In the context of user experience improvements, the addition of more detailed guidelines, integration of sound effects and implementation of achievement badges or reward system have been proposed.
- Within the scope of educational developments, it was emphasized that physics and mathematics problems should be included for sensor usage simulation, robotic kit integration and STEM integration.

When the overall evaluation of the research findings is made, it is seen that LRwtG game largely meets the goal of teaching robotic coding to children ages 6-12. Especially in terms of the potential of problem solving and developing algorithmic thinking skills, the game has been found to be very successful. The game's educational value and user experience have also been evaluated quite positively. The survey results confirm the stated advantages of the Scratch-based maze game in the literature: low cost, high availability, effective problem-solving focus and high student motivation. Although improvements can be made in terms of visual design and technical features, it is envisaged that the effectiveness of the game can be further increased by developing the level system, programming features and user experience in line with the recommendations of the participants. These findings suggest that

Scratch-based educational games are an effective tool in robotic coding education, and that such approaches should be disseminated in future studies.

CONCLUSION AND SUGGESTIONS

The LRwtG game developed in this study is an educational game developed using Scratch to teach robot programming to children aged 6-12. Before the game was developed, relevant studies in the literature, current games and related theses were examined, then the developed game was presented to expert educators in the field and the opinions of the participants were taken through questionnaire. The results are supported by studies in the literature.

The design, which aims to provide robot programming training to young age groups, has adopted the approach of creating a learning game environment at its own pace. This flexibility allows them to adapt to different learning speeds (Vygotsky, 1978). Both academic and commercial platforms recognize that block-based programming is appropriate for younger age groups (Strawhacker et al., 2018; Aytekin et al., 2018). As the game developed appeals to young age groups, a block-based programming-based strategy is used in the design of the game to teach robot coding. The focus on problem-solving and algorithmic thinking in design is aligned with the educational priorities (Çetin & Demircan, 2018).

The applied survey provided valuable insights into the educational value and effectiveness of LRwtG gaming. The results show that the game has significant educational potential, especially in developing problem-solving and algorithmic thinking skills. Its positive impact on algorithmic thinking skills is supported by studies in the literature. The researchers note that one of the educational benefits of academic studies on Scratch and robotic coding is the development of children's algorithmic thinking and problem-solving skills (Yilmaz & Kara, 2023; Karataş, 2021; Tekin & Keser, 2020).

User experience aspects of the game have received positive feedback due to its user-friendly interface that makes learning easier. Academic research shows that block-based programming tools like Scratch facilitate coding learning and support creative thinking skills (Canbekek, 2020; Salahli et al., 2019). The game's level-based progression system has been particularly effective at maintaining user engagement and motivation while gradually introducing more complex coding concepts. The literature also shows that robotics coding activities increase student motivation and interest in the classroom (Tekin & Keser, 2020; Puritat, 2019). The recommendation rate among participants of the survey is particularly encouraging, with the majority stating that they would recommend the game for educational use.

Scratch's visual and block-based structure may be inadequate for the development of advanced programming skills, making the transition to text-based programming languages difficult (Aytekin et al., 2018; Strawhacker et al., 2018). The cost of hardware and software required for robotic coding activities can be a barrier for schools, especially in disadvantaged areas (Indelete & Land, 2023; Ejiwale, 2013). Some research has also shown that teachers may not have enough knowledge and experience in implementing robotic coding activities, which can negatively impact the efficiency of these activities (Hunter & Succeedan, 2021; Kanmaz, 2023). Also, the typically short-term nature of Scratch events makes it difficult to assess long-term effects (Tekin & Keser, 2020; Canbeldek, 2020).

As a result, it can be said that LRwtG game developed can be engaging and effective in teaching young students the concept of robot coding. The simplicity and user-friendly interface of the game supports efficient learning for children aged 6-12 (Strawhacker et al., 2018). The game environment, which can be accessed without the need for special equipment, is considered to provide focus and motivation for this age group (Salahli et al., 2019). On the other hand, it is undeniable that content needs to be improved and more advanced visual design should be provided in order to increase the educational impact of the game.

Recommendations for Future Work

Based on the findings of this study, the following recommendations for future research and development activities are presented:

Content Development and Expansion: More advanced programming concepts can be integrated while maintaining the current format of the game. Incorporating basic programming structures such as variable use, loop structures and conditional statements into the game mechanism can enable students to gain extensive programming experience.

Platform Diversity and Accessibility: The game can be make playable on different devices and operating systems. The development of optimized versions for mobile devices, tablet computers, and web-based platforms will expand the scope of access.

Adaptive Learning Systems: Implementation of systems that automatically adjust the difficulty level according to individual user performance can be made. This approach will allow each student to progress at his or her own pace.

Teacher Support Systems: Comprehensive teacher guides and support materials can be developed for effective use in the classroom environment. These resources will ensure that the game is used in accordance with the curriculum objectives.

Competitive Learning Environments: It is recommended to add social learning elements such as multiplayer modes and leadership tables. These features will increase student motivation and promote collaborative learning.

User Interface Optimization: The game interface in terms of user experience can be improved. Improving visual effects and improving the quality of interaction will improve the overall gaming experience.

Motivation-Boosting Strategies: Success badges, progress tracking systems and personalized feedback mechanisms can be integrated. These elements will play a critical role in maintaining student motivation.

The game developed in this study has the potential to make robotic coding education accessible. The studies to be carried out in line with the proposed developments will significantly increase the effectiveness of educational games in robotic coding education.

REFERENCES

Akgün, S. (2020). [The Effect Of Scratch And Robotics Applications On Computational Thinking Skills Of Middle School Students].

Alan, D. (2017). Dijital Oyun Tabanlı Yaklaşım Ile Yazılım Geliştirme Öğretimi [Yüksek Lisans Tezi]. Selçuk Universitesi, Fen Bilimleri Enstitüsü.

Alper, A., & Şanlı, M. (2024). Kodlama Ve Robotik Eğitimi Alanında Türkiye'de Yapılan Lisansüstü Tezlerin İncelenmesi. Ankara University Journal of Faculty of Educational Sciences (JFES), 57(2), 759-803.

Andrias, R. M., Sunar, M. S., & Sondoh Jr, S. L. (2021, December). Adaptive Gamification: User/Player Type And Game Elements Mapping. In International Conference On Intelligent Technologies For Interactive Entertainment (Pp. 242-256). Cham: Springer International Publishing.

Arslan, K. (2022). [Micro:Bit Ile Scratch 3].

Arslan, U. (2023). Çocuklar Için Ücretsiz 10 Kodlama Oyunu. Retrieved: January 09, 2023 Https://Okul.Com.Tr/Aileler-Icin/Makaleleri/Cocuklar-Icin-Ucretsiz-10-Kodlama-Oyunu-562

Attila, K., & Szilvia, S. (2022). From Scratch To Python-Lego Robots As Motivational Tools For Coding. Multidiszciplináris Tudományok, 12(3), 247-255.

Avci, Ü., & Başaran, R. (2021). Robotik Kodlama Eğitmeni Eğitiminden Yansımalar: Öğretmen Görüşleri. Eğitim Ve Psikoloji Üzerine Araştırmalar, 5(2), 223–245. Https://Doi.Org/10.54535/Rep.1007946

Aytekin, A., Çakır, F. S., Yücel, Y. B., & Kulaözü, İ. (2018). Geleceğe Yön Veren Kodlama Bilimi Ve Kodlama Öğrenmede Kullanılabilecek Bazı Yöntemler. Avrasya Sosyal Ve Ekonomi Araştırmaları Dergisi, 5(5), 24-41.

Barradas, R., Lencastre, J. A., Soares, S., & Valente, A. (2019). Usability Evaluation Of An Educational Robot For STEM Areas. In Proceedings Of The 11th International Conference On Computer Supported Education (Pp. 218-225).

Bevz, M. (2023). Exploring the Use of Color as an Informational Layer in Video Game Design. *PQDT-Global*.

Blockly Games. (2024). Access Date: 2024, October 11 , Https://Blockly.Games/?Lang=En

Breiner, J. M., Johnson, S. S., Harkness, C. C., & Koehler, C. M. (2012). What Is STEM? A Discussion About Conceptions Of STEM In Education And Partnerships. School Science And Mathematics, 112(1), 3–11.

Bugmann, J., & Karsenti, T. (2018). Learning to program a humanoid robot: impact on special education students. Research on e-Learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives, 323-337.

Burchinal, M., Vandergrift, N., Pianta, R., & Mashburn, A. (2010). Threshold Analysis Of Association Between Childcare Quality And Child Outcomes For Low-Income Children In Pre-Kindergarten Programs. Early Childhood Research Quarterly, 25(2), 166-176.

Canbeldek, M. (2020). Erken Çocukluk Eğitiminde Üreten Çocuklar Kodlama Ve Robotik Eğitim Programının Etkilerinin Incelenmesi [Doktora Tezi]. Pamukkale Üniversitesi Eğitim Bilimleri Enstitüsü, Temel Eğitim Anabilim Dalı, Okul Öncesi Eğitimi Bilimi.

Code Combat. (2024). Codecombat Game Platform. Retrieved November 21, 2024, From Https://Codecombat.Com/Play

Code Karts. (2023). Retrieved: 2023, August 31. Https://Www.Gameloop.Com/Tr/Game/Education/Code-Karts-Pre-Coding-For-Kids-On-Pc

Code.Org. (2024, November). About Code.org. https://code.org/about

Code.Org. (2024-a). Pre-Express Course Lesson Retrieved November 21, 2024, From Https://Studio.Code.Org/S/Pre-Express-2024/Lessons/

Codemonkey Studios Inc. (2024, August 8). Coding For Kids. Https://Www.Codemonkey.Com/

Codingame. (2024). Onboarding Puzzle. Retrieved August 15, 2024, From Https://Www.Codingame.Com/Ide/Puzzle/Onboarding

Coşkunserçe, O. (2021). Eğitimde Robot Programlama. Pegem Akademi.

Crunchzilla. (2024). Code Monster. Retrieved August 20, 2024, From Https://Www.Crunchzilla.Com/Code-Monster

Çengelci, B., & Çimen, H. (2005). Endüstriyel Robotlar. Makine Teknolojileri Elektronik Dergisi, 2(2), 69-78.

Çetin, M., & Demircan, H. Ö. (2018). Robot Programlama Yoluyla STEM Eğitimi Için Teknoloji Ve Mühendisliğin Güçlendirilmesi: Sistematik Bir Literatür Taraması. Erken Çocuk Gelişimi Ve Bakımı, 190(9), 1323–1335.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements To Gamefulness: Defining "Gamification". Proceedings Of The 15th International Academic Mindtrek Conference: Envisioning Future Media Environments, 9-15.

Dizman, A. (2018). Kodlama, Robotik, 3d Tasarım Ve Oyun Tasarımı Eğitiminin 11-14 Yaş Grubu Öğrencilerinin Problem Çözme Becerileri Ve Üstbilişsel Farkındalık Düzeyine Etkisi [Yüksek Lisans Tezi]. Bahçeşehir Universitesi, Eğitim Bilimleri Enstitüsü.

Eğin, F. (2019). Bilişim Teknolojileri Öğretmenlerinin Kodlama Öğretimine Ilişkin Görüşlerinin Incelenmesi [Yüksek Lisans Tezi]. Ege Universitesi, Fen Bilimleri Enstitüsü.

Ejiwale, J. (2013). Barriers To Successful Implementation Of STEM Education. Journal Of Education And Learning, 7(2), 63–74.

Eraytaç, Ö. F. (2019). Robotik Kodlama Eğitiminde Blok Tabanlı Kodlama Yönteminin Ortaokul Öğrencilerinin Akademik Başarısına Etkisi [Yüksek Lisans Tezi]. Çukurova Universitesi, Sosyal Bilimler Enstitüsü.

Erdel,E. (2024). İlkokul Öğrencilerine Uygulanan Bilgisayarsiz Bilgisayar Bilimi, Blok Tabanli Ve Robotik Kodlama Eğitiminin, Öğrencilerin Bilgi Işlemsel Düşünme Ve Kodlama Becerilerine Etkilerinin Incelenmesi Yüksek

Lisans Tezi]. Bolu Abant Izzet Baysal Universitesi, Lisansüstü Eğitim Enstitüsü.

Fidan, A. (2016). Scratch ile Programlama Öğretiminde Oyunlaştırmanın Öğrenci Katılımına Etkisi [Yüksek Lisans Tezi]. Bursa Uludag University, Eğitim Bilimleri Enstitüsü.

Foldager, N., Hansen, H., Tewes, M. S., & Bjørner, T. (2017). Designing An Engaging And Informative Application About First Aid: Gamification And Humor As Design Elements In A Serious Game. In Smart Objects And Technologies For Social Good: Second International Conference, GOODTECHS 2016 (Pp. 78-87). Springer International Publishing.

Gameloop. (2023). Code Karts: Pre-Coding For Kids On PC. Retrieved From Https://Www.Gameloop.Com/Tr/Game/Education/Code-Karts-Pre-Coding-For-Kids-On-Pc

Gil-Aciron, L. A. (2024). The gamer psychology: a psychological perspective on game design and gamification. Interactive learning environments, 32(1), 183-207.

Haymana, İ., & Özalp, D. (2020). Robotik Ve Kodlama Eğitiminin Ilkokul 4. Sınıf Öğrencilerinin Yaratıcı Düşünme Becerilerine Etkisi. İstanbul Aydın Üniversitesi Eğitim Fakültesi Dergisi, 6(2), 247-274.

Ibrahim, R., & Jaafar, A. (2009). Eğitsel Oyunlar (EG) Tasarım Çerçevesi: Oyun Tasarımı, Pedagoji Ve Içerik Modellemesinin Birleşimi. 2006, 293–298. Https://Doi.Org/10.1109/Iceei.2009.5254771

Jet Brains. (2024, September 30). Robocode. Https://Robocode.Sourceforge.Io/

Kanmaz, T. (2023). Okul Öncesi Öğretmenleri Robotik Kodlama Hakkında Ne Düşünüyor. Dumlupınar Üniversitesi Eğitim Bilimleri Enstitüsü Dergisi, 7(1), 1-22.

Karataş, H. (2021). 21. Yüzyıl Becerilerinden Robotik Ve Kodlama Eğitiminin Türkiye Ve Dünyadaki Yeri. 21. Yüzyılda Eğitim Ve Toplum Eğitim Bilimleri Ve Sosyal Araştırmalar Dergisi, 10(30), 693-729.

Kaya, M., Korkmaz, Ö., & Çakır, R. (2020). Oyunlaştırılmış Robot Etkinliklerinin Ortaokul Öğrencilerinin Problem Çözme Ve Bilgi Işlemsel Düşünme Becerilerine Etkisi. Ege Eğitim Dergisi, 21(1), 54-70.

Kılınç, S. (2017, April 10). Kodlamayı Eğlenceli Yollarla Öğrenmenizi Sağlayan 15 Ücretsiz Oyun! Https://Www.Webtekno.Com/Kodlamayi-Eglenceli-Yollarla-Ogrenmenizi-Saglayan-15-Ucretsiz-Oyun-H27675.Html

Kocaçil, S. (2020). Programlanabilir Eğitsel Oyuncak Robot Tasarımı Ve Okul Öncesi Dönemde Kullanılması. [Yüksek Lisans Tezi]. Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Anabilim Dalı.

Kodable. (2024). Kodable Demo. Retrieved November 24, 2024, From Https://Dashboard.Kodable.Com/Demo

Kodable. (2024-a). Kodable Basics Learning Guide. Http://Resources.Kodable.Com/Public/Materials/KB.Pdf

Konyaoğlu, C. (2019). Robotik Kodlama Eğitiminin Ortaokul Öğrencilerinin Problem Çözme Becerilerine Etkileri Ve Öğrencilerin Robotik Kodlama Etkinliklerine Ilişkin Görüşleri [Yüksek Lisans Tezi] Bolu Abant İzzet Baysal University, Eğitim Bilimleri Enstitüsü.

Marcus, A. (2002). Graphic Design For Electronic Documents And User Interfaces. ACM Press.

Mattingly, Jon. (2024, September). All Collections- Learning Outcomes- What Courses Does Kodable Basics Offer? Http://Support.Kodable.Com/En/Articles/5650473-What-Courses-Does-Kodable-Basics-Offer

Munsell, A. H. (1912). A Color Notation. Munsell Color Company.

Nolen, S. B., & Koretsky, M. D. (2018). Affordances of virtual and physical laboratory projects for instructional design: Impacts on student engagement. *IEEE Transactions on Education*, *61*(3), 226-233.

Oliver, J. S. (2018). The Implementation Of Integrated Science Technology, Engineering And Mathematics (STEM) Instruction Using Robotics In The Middle School Science Classroom.

Özdemir, M. (2023). Turkish Teachers' Gamification User Types And Preferences Of Game Elements For Their Instruction. [Master's Thesis]. Middle East Technical University, Computer Education and Instructional Technologies.

Özgür Şeremet. (2024, October 1). Scratch Örnekleri 14 - Labirent. Https://Ozgurseremet.Com/Scratch-Ornekleri-14-Labirent/

Pacheco, H., & Macedo, N. (2021). ROSY: An Elegant Language To Teach The Pure Reactive Nature Of Robot Programming. In 2020 Fourth IEEE International Conference On Robotic Computing (IRC) (Pp. 240-247). IEEE.

Piaget, J. (1952). The Origins Of Intelligence In Children. Norton.

Puritat, K. (2019). Enhanced Knowledge And Engagement Of Students Through The Gamification Concept Of Game Elements. International Journal Of Engineering Pedagogy, 9(5).

Robocode Home Page. (2024). Sourceforge. Retrieved September 30, 2024, From Https://Robocode.Sourceforge.Io/

Robocode. (2023, October 24). Https://Robocode.Sourceforge.Io/

Şahin, H., & Arıkan, A. (2024). Okul Öncesi Eğitimde Robotik Uygulamaları. Trakya Eğitim Dergisi, 14(1), 260-286.

Şahin, M. (2021). Çocuk Gelişimi Programı Öğrencilerinin Okul Öncesi Dönem Çocuklarına Yönelik STEM Temelli Hazırlanan Etkinlikler Ile Ilgili Görüşleri. Ahi Evran Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 7(2), 518-536.

Salahli, M. A., Gasimzadeh, T., Alasgarova, F., & Guliyev, A. (2019). A Mobile Application To Teach Coding And Computational Thinking In Scratch Programming Language. In 13th International Conference On Theory And Application Of Fuzzy Systems And Soft Computing—ICAFS-2018 (Pp. 807-814). Springer International Publishing.

Salen, K., & Zimmerman, E. (2003, January). This Is Not A Game: Play In Cultural Environments. In Proceedings Of Digra 2003 Conference: Level Up.

Silvera-Tawil, D. (2024). Robotics In Healthcare: A Survey. SN Computer Science, 5(1), 189.

Sourceforge. (2024). Robocode Project Page. Retrieved October 4, 2024, From Https://Sourceforge.Net/Projects/Robocode/

Strawhacker, A., Lee, M., & Bers, M. U. (2018). Teaching Tools, Teachers' Rules: Exploring The Impact Of Teaching Styles On Young Children's Programming Knowledge In Scratchjr. International Journal Of Technology And Design Education, 28, 347-376.

Tasdemir, Ş., Çizmeci, H. İ., & Alan, D. (2016). Learning Mathematics With Educational Digital Game Programming. Research Highlights In Education And Science, 2016, 149-155.

Tekin, Y., & Keser, H. (2020). Matematik Öğretiminde Robotik Etkinliklerin Öğrencilerin Derse Yönelik Güdülenme, Tutum Ve Başarılarına Etkisi Ve Bir Eğitim Ortamı Önerisi. Akdeniz Eğitim Araştırmaları Dergisi, 14(34), 472-493.

Tekinbas, K. S., & Zimmerman, E. (2003). Rules Of Play: Game Design Fundamentals. MIT Press.

Thanyaphongphat, J., Thongkoo, K., & Daungcharone, K. (2023, March 22). A Micropython-Based Educational Robotic Maze Approach: Learning Improvement To Competition. Https://Doi.Org/10.1109/Ectidamtncon57770.2023.10139363

Totan, H. N. (2021). Blok Tabanlı Kodlama Eğitiminin Ortaokul Öğrencilerinin Bilgi Işlemsel Düşünme Becerileri Ve Kodlama Öğrenimine Yönelik Tutumlarına Etkisi: Blocky Örneği [Master's Thesis]. Necmettin Erbakan Üniversitesi, Eğitim Bilimleri Enstitüsü.

Tynker. (2024). Tynker: Coding For Kids. Retrieved October 11, 2024, From Https://Www.Tynker.Com/

Vygotsky, L. S. (1978). Mind In Society: The Development Of Higher Psychological Processes. Harvard University Press.

Xiao, B., Chen, C., & Yin, X. (2022). Recent Advancements Of Robotics In Construction. Automation In Construction, 144, 104591.

Yaman, S. (2023). Hikayeleştirilmiş Robotik Kodlama Programlarının Okul Öncesi Çocuklarının Sosyal-Duygusal Becerilerine Etkisi. [Yüksek Lisans Tezi]. İstanbul Aydın Üniversitesi Lisansüstü Eğitim Enstitüsü.

Yılmaz, F. (2018). Robotlar Hayatımızda. FSM İlmî Araştırmalar İnsan Ve Toplum Bilimleri Dergisi, 12, 109-120.

Yılmaz, G. (2023). Okul Öncesi Eğitimde STEM Yaklaşımının Kullanımı: Oyun Temelli Mühendislik Tasarım Uygulamaları. [Doktora Tezi]. Bursa Uludağ Üniversitesi. Eğitim Bilimleri Enstitüsü

Yılmaz, H. E., & Kara, M. (2023). Eğitimde Robot Kullanımının Avantajları: Ulusal Tezlerin Yöntemleri Ve Bulguları Üzerine Bir Inceleme. Bilgi Ve İletişim Teknolojileri Dergisi, 5(2), 115-129.

Yukselturk, E., & Altıok, S. (2016). BT Öğretmen Adayları Tarafından Scratch Görsel Programlama Aracı Ile Geliştirilen Eğitsel Oyunların İncelenmesi. SDU International Journal Of Educational Studies, 3(1), 59-66.