**ATILIM UNIVERSITY**

**FACULTY OF ENGINEERING**

**ISE/SE 494 Senior Project Report**

**MATHFUN**

**PREPARED BY**

**ARMİN ALİ**

**HASAN EMREHAN TOPÇU**

**HÜSEYIN BERKE ÜNAL**

**SUPERVISOR**

**ASSOC. DR. CANSU ÇİĞDEM EKİN**

**2022-23 Fall**

**Abstract**

The value of education is constantly growing in the world today. However, as education levels rise, learning issues become much more prevalent. Students' motivation to learn decreases along with these difficulties. It was thought how to solve this problem and it was decided to develop a game about learning mathematics, which is a difficult lesson. The subject of permutation-combination was chosen in the mathematics lesson. The main purpose here is to ensure that the student learns the subject of the course effectively and does not experience low motivation while learning. C # and Unity program were used as programming language during the production of the game. The figures used in the game were also taken ready-made from the internet. A design was made that could attract the attention of high school students and provide effective learning. With this game, it is thought that students will learn the topic they have chosen efficiently.

Keywords: Teens, High School, Permutation-Combination, Mobile Application, Educational Game.

**Acknowledgements**

We offer our respect and thanks to our esteemed teacher Cansu Çiğdem Ekin, who always supported us in all the problems we encountered during the research and development process, answered our simplest questions with patience and a smiling face, and guided us with all his knowledge and experience.

**Table of Content**

[**Abstract** i](#_Toc123672594)

[**Acknowledgements** ii](#_Toc123672595)

[**List of Figures** v](#_Toc123672596)

[**List of Tables** vii](#_Toc123672597)

[Ch1: Introduction 1](#_Toc123672598)

[1.1. Problem Statement 1](#_Toc123672599)

[1.2. Purpose 1](#_Toc123672600)

[1.3. Scope 2](#_Toc123672601)

[1.4. Definitions, Acronyms, and Abbreviations 2](#_Toc123672602)

[Ch2: Project Plan 3](#_Toc123672603)

[2.1. Methodology 3](#_Toc123672604)

[2.2. Time Schedule 3](#_Toc123672605)

[2.3. Deliverables List 4](#_Toc123672606)

[Ch3: Literature Review 5](#_Toc123672607)

[Ch4: Requirements 7](#_Toc123672608)

[4.1. Overall Description of the Project 7](#_Toc123672609)

[4.1.1. Product Perspective 7](#_Toc123672610)

[4.1.2. Product Functions 8](#_Toc123672611)

[4.1.3. User Characteristics 8](#_Toc123672612)

[4.1.4. Constraints 9](#_Toc123672613)

[4.1.5. Assumptions and dependencies 9](#_Toc123672614)

[4.2. Specific Requirements 9](#_Toc123672615)

[4.2.1. Functional Requirements 9](#_Toc123672616)

[4.2.2. Non-functional Requirements 9](#_Toc123672617)

[4.2.3. Specific Requirements 10](#_Toc123672618)

[Ch5: Design 13](#_Toc123672619)

[5.1. Usability of the Design 13](#_Toc123672620)

[5.2. Diagram Design 13](#_Toc123672621)

[5.2.1. Class Diagram of the System 13](#_Toc123672622)

[5.2.2. Use Case Diagram 14](#_Toc123672623)

[5.2.3. User Interface Design 15](#_Toc123672624)

[5.2.4. Software Development Methodology 19](#_Toc123672625)

[5.2.5. Timeline 20](#_Toc123672626)

[5.2.6. Budget and Cost 21](#_Toc123672627)

[5.2.7. Collaboration 23](#_Toc123672628)

[Ch6: Implementation & Test 24](#_Toc123672629)

[6.1 Implementation 24](#_Toc123672630)

[6.2. Test 40](#_Toc123672631)

[Ch7: Evaluation 43](#_Toc123672632)

[Ch8: Impact of the Project & Compliance with the Constraints 44](#_Toc123672633)

[8.1. Compliance with the Realistic Constraints 44](#_Toc123672634)

[8.2. Impact of the Project 45](#_Toc123672635)

[Ch9: Conclusion 49](#_Toc123672636)

[References 50](#_Toc123672637)

[Appendix 52](#_Toc123672638)

**List of Figures**

[Figure 1. Difficulty index (Durmus, 2004). 5](#_Toc123162555)

[Figure 2. Start screen. 7](#_Toc123162556)

[Figure 3. Use case diagram. 11](#_Toc123162557)

[Figure 4. ER diagram. 12](#_Toc123162558)

[Figure 5. Class diagram. 13](#_Toc123162559)

[Figure 6. Use case diagram. 14](#_Toc123162560)

[Figure 7. Start page. 15](#_Toc123162561)

[Figure 8. First scene. 16](#_Toc123162562)

[Figure 9. First tutorial. 17](#_Toc123162563)

[Figure 10. Question part. 18](#_Toc123162564)

[Figure 11. Collecting stars. 18](#_Toc123162565)

[Figure 12. Spiral model. 20](#_Toc123162566)

[Figure 13. Timeline table. 21](#_Toc123162567)

[Figure 14. Cost table. 22](#_Toc123162568)

[Figure 15. Collaboration diagram. 23](#_Toc123162569)

[Figure 16. Pause & resume button. 24](#_Toc123162570)

[Figure 17. Pause function. 25](#_Toc123162571)

[Figure 18. Game stop screen. 25](#_Toc123162572)

[Figure 19. Resume function. 26](#_Toc123162573)

[Figure 20. Pause & resume button onClick. 26](#_Toc123162574)

[Figure 21. Play button. 26](#_Toc123162575)

[Figure 22. Play function. 27](#_Toc123162576)

[Figure 23. Start screen. 27](#_Toc123162577)

[Figure 24. Gameplay. 28](#_Toc123162578)

[Figure 25. Play button onClick. 28](#_Toc123162579)

[Figure 26. First hint onClick. 29](#_Toc123162580)

[Figure 27. Coroutine function call. 29](#_Toc123162581)

[Figure 28. Coroutine function. 29](#_Toc123162582)

[Figure 29. First hint screen. 30](#_Toc123162583)

[Figure 30. Continue button function. 30](#_Toc123162584)

[Figure 31. Continue button. 30](#_Toc123162585)

[Figure 32. Hint button. 31](#_Toc123162586)

[Figure 33. Hint function. 31](#_Toc123162587)

[Figure 34. Back to menu option. 31](#_Toc123162588)

[Figure 35. Back to menu function. 31](#_Toc123162589)

[Figure 36. Hint screen. 32](#_Toc123162590)

[Figure 37. The function of hint’s movements. 32](#_Toc123162591)

[Figure 38. Hints onClick. 33](#_Toc123162592)

[Figure 39. Character movements. 33](#_Toc123162593)

[Figure 40. Animation. 33](#_Toc123162594)

[Figure 41. Character movements function. 34](#_Toc123162595)

[Figure 42. Handle movement function. 34](#_Toc123162596)

[Figure 43. Gameplay. 35](#_Toc123162597)

[Figure 44. Gameplay. 35](#_Toc123162598)

[Figure 45. Colliders function. 36](#_Toc123162599)

[Figure 46. Gameplay. 37](#_Toc123162600)

[Figure 47. Question screen. 37](#_Toc123162601)

[Figure 48. Level complete screen. 38](#_Toc123162602)

[Figure 49. Stars’ inspector. 38](#_Toc123162603)

[Figure 50. Stars’ image. 39](#_Toc123162604)

[Figure 51. Questions’ inspector. 39](#_Toc123162605)

[Figure 52. Question image. 40](#_Toc123162606)

**List of Tables**

[Table 1. Gantt chart. 3](#_Toc123162608)

[Table 2. Realistic constraints & condition. 44](#_Toc123162609)

[Table 3. Impact assessment report. 46](#_Toc123162610)

Ch1: Introduction

* 1. Problem Statement

Today, with the development of technology, the use of technological tools has become widespread. However, people began to spend more time with technological devices, to meet their needs with these tools and to do their daily work. In addition, today's students have started using technology to discover new skills and learn their lessons, and they have seen many benefits. In both academic and corporate environments, education is crucial. In the former, workers are assisted in changing their previous behaviors by means of education or training. The goal of instruction in the latter is to pique pupils' curiosity. Using technology can improve students' understanding and memory of concepts in either situation. Technology has certain advantages in the field of education. These: It increases kids' motivation to learn, give busy students the option to work at home on their own schedules, teach pupils new technological abilities they can apply in the workplace later, lower paper and photocopy expenses while promoting the "green revolution" idea (Kumar KC, 2018). However, when we look at the situation in our country, we see that this situation is not very valid. The success rate at all levels is very low and students do not use technological resources correctly. In this case, it causes failure especially in numerical courses. According to the article titled “Gerçeklerle Yüzleşme” researched by Bütüner and Güler (2017), a significant decrease was observed in mathematics scores in Turkey between 1999 and 2015. For this reason, Turkey remained below the international average.

On the other hand, during the Covid-19 process, students continued their lessons online and the education process was moved to the technological platform.

* 1. Purpose

The main purpose of this project is to create a platform that will facilitate learning permutation-combination topics, which are difficult for high school students, to learn by making them fun. This platform will also include the method of learning through games, which is a way to ensure active participation of students and learning with pleasure due to its visual elements. In this way, it is foreseen that education and training will be moved to technological platforms and this will be supported, especially after Covid-19. With the planned project, it is thought that this deficiency will be eliminated.

In this report is to clearly state all the requirements, information about the game we will release and to indicate all the ways we will follow functionally in this process. Thus, we specify what kind of features and requirements the system will have such as functional and non-functional requirements, use case diagrams, overall designs.

* 1. Scope

The scope of the MF (Math Fun) system is to teach mathematics subjects that high school students have difficulty in learning while entertaining them. It deals with the subject systematically and puts it in the mind of the student with easy methods. In addition, it provides motivation at every stage by informing the student about the current progress and the current situation.

* 1. Definitions, Acronyms, and Abbreviations
* Definitions

Curriculum: That is a standards-based sequence of planned experiences where students practice and achieve proficiency in content and applied learning skills. Curriculum is the central guide for all educators as to what is essential for teaching and learning, so that every student has access to rigorous academic experiences (RIDE).

* Acronyms

MF: Math Fun

UC: Use Case

SRS: Software Requirements Specification.

Ch2: Project Plan

Our project includes quantitative research. We analyzed the necessary information, research, and numerical data about the problem we identified. As a result, we decided to follow the deductive logic.

* 1. Methodology

In our project, we decided to use the waterfall methodology because we thought that we would proceed in a linear and one-way manner. The reason for this is because we want to progress step by step within certain limits and the project requested from our group is clear and there is no question mark.

* 1. Time Schedule

Table 1. Gantt chart.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task No | Task Name | Start Date | End Date | Weeks | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | Building the basics of the game | 06.10.22 | 18.10.22 | X | X |  |  |  |  |  |  |  |  |  |
| 2 | Creation of the game interface | 19.10.22 | 10.11.22 |  |  | X | X | X |  |  |  |  |  |  |
| 3 | Repair of bugs and errors in the interface | 10.11.22 | 02.12.22 |  |  |  |  |  | X | X | X |  |  |  |
| 4 | Adding details with the game | 03.12.22 | 22.12.22 |  |  |  |  |  |  |  |  | X | X | X |

* 1. Deliverables List

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Type | Description | Due Date |
| Project Proposal | Word | General information about our project | 26.08.2022 |
| Introduction  Literature Review  Project Plan  SRS | Word | Our detailed explanation of our project, the information we have obtained because of our research, our time planning, and requirements | 20.10.2022 |
| Prepare a Intermediate Report | Word | Preparing the design and implementation part of our report | 25.11.2022 |
| Final Version of the Report | Word | We have completed and delivered the remaining parts of the report. | 28.12.2022 |

Ch3: Literature Review

We started our project by researching the necessary resources and supported our article with these researches. Mathematics is a field that is not easy to learn because it is an abstract discipline by nature (Kücük, Topcu & Göktas, 2014). It becomes even more difficult for today's students, especially in certain subjects. Courses in probability, permutation, and combination are directly related to the fact that probability has started to be taught in public schools, have a direct bearing on daily life and may be related indirectly or directly to other disciplines. They also play an important role in the probabilistic reasoning that we commonly use. Probability, permutation, and combination are crucial concepts, although these are also among the hardest to comprehend and teach because of how important they are. In this situation, issues develop (Gün & Others, 2021). In the table below, math topics are given with difficulty indexes. When we look at this table, we see that the subject of permutation-combination has a high difficulty index. We chose the subject of permutation-combination among these subjects.

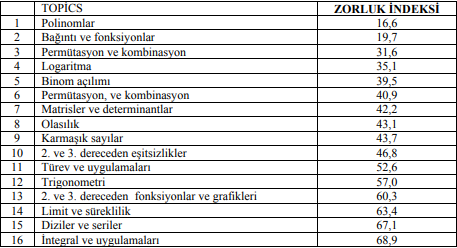


Figure 1. Difficulty index (Durmus, 2004).

Teaching probability, permutation and combination subjects through games based on the tests will make an important contribution towards the learning purpose. The student, who is both active and learning by having fun in the process, will now learn by enjoying the mathematics that he has been afraid of since his childhood. In addition, the fact that the subjects that the students have difficulty with become enjoyable and the logic of the subject will be internalized and learned by discovering it instead of being theoretical will both make the subject permanent learning and raise qualified individuals that individuals should have. Students will overcome their misconceptions through active learning, and they will gain high-level thinking skills in the process (Gün & Others, 2021).

As can be seen in the table, the subject we chose is among the subjects that students in our country have the most difficulty with. That's why we thought of a way to teach students the subjects they have difficulty with in an effective and fun way, and we decided to develop a game by embodying this subject. Today, students' learning methods have changed, and they are learning a lot from games. As a result, we conclude that games have an educational and instructive aspect. In general, games teach subjects that are difficult to understand and are used as effective educational tools in understanding (Chizary & Farhangi, 2017).

Math education can also be accomplished through educational video games. Mathematical educational video games are utilized in the classroom to inspire students, alter their misconceptions about the subject, assist them in overcoming their math phobia, and dispel any preconceived ideas they might well have (Kebritchi, Hirumi & Bai, 2010). The benefits of using educational video games in the teaching of mathematics to students have been the focus of numerous research. The majority of these studies claim that instructional computer games improve motivation and achievement (Lopez-Morteo & Lopez, 2007).

In the study, students interacted more with each other when they were at the computer or mobile. Students' interactions with each other differed from those in the traditional classroom, and behaviors such as helping each other instantly, wondering what each other was doing, and getting excited were observed. In the projects carried out, the students were found to be at a high level as a result of long-term education (Mercan & Others, 2009).

In line with the literature research we have done, we are planning to design a video game that students can easily access, learn efficiently and not get bored while learning.

Ch4: Requirements

* 1. Overall Description of the Project

### Product Perspective

The game we will make in our project basically aims to increase the rate of students' learning about the subject of permutation and combination. In addition, it is aimed to ensure that they have fun while learning and to provide stability while working. The application proceeds step by step, gives information about the subject, and then moves on to the gameplay part of the game. The student is asked questions related to the topic we have determined. When the student finishes the level, the student has chosen, he/she is scored based on student’s answers. If the student wishes, he can continue the game by getting hints about the next level.

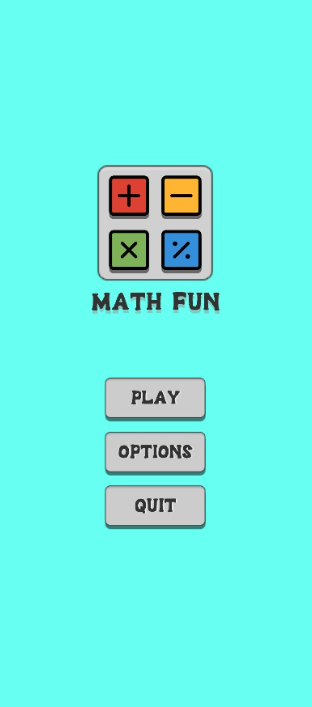


Figure 2. Start screen.

### Product Functions

The application will have the main listed features:

* Log into the game.
* Look for course topics you want to study.
* Watch the tutorial.
* Start the game.
* Choose the level to start.
* Start the game with your game character.
* Answer the questions on the obstacles that come your way.
* Complete the section.
* See your score.
* Go to the next level.

### User Characteristics

There are two main users in our game since it is mobile runner game. The first user is the main user that plays the game which is the students. The second user we can consider as the producers that control the dynamic of the game.

User (Student): Student can be said that is the main user of the program. He/She controls the player as He/ She starts the game. Student tries to avoid obstacles and answer the questions correctly to pass every level. Student’s main actions are the movement of the player and making the mathematical operations.

Producers: Producer is mainly responsible for the trend of the game. The student should analyze every step of the game before, during, and after the gaming process for the sustainability. Producer must resolve every bug before the production or should solve it in momentarily bugs and errors. The student also is responsible for flow of the main scenario. We mean that, producers should innovate the game once a while for better experience and to avoid monotony.

### Constraints

1. The whole systems structure should conform the Process Impact Intranet Development Standard.
2. The game should be structured in unity for playability in mobile platforms.
3. All scripts should be written in C# by using visual studio code or XCode.
4. All designs should be visible in Unity platform.

### Assumptions and dependencies

* The system constantly keeps track of the student's score.
* The student must know enough questions and collect points to pass the stages. Then you can move on to the next step.
  1. Specific Requirements

### Functional Requirements

* The game shall train the given data with necessary algorithms.
* The game shall recognize screen touching mechanics.
* The game should understand the achievement of game levels.
* The game should shuffle the questions specified by levels.
* The game should provide solutions of each question.
* The game should provide guides of topics.

### Non-functional Requirements

* The response time of character by players screen touching.
* The Game must stop when player crashes to an obstacle or wrong answers to a question.
* The game must increase the level as the player answers set of questions right.
* The game must respond with a sound when player crashes to an obstacle.
* The game must store every student’s login information.
* The game must increase the game of speed as the level increase.
* The game be reliable with the specific mathematic topics.
* The game must be up to date with the relevant permutation and combination topics.
* The game must recognize the player’s name and should store the records.

### Specific Requirements

* The game must pop up a question after player dodges an obstacle.
* The game must show topic introduction of permutation and combination before each relevant level.
* The game must show the health bar of the player after each crash.
* The game must display each navigation according to accomplishments.
* The game must give a greeting message and a specific sound after finishing every level.

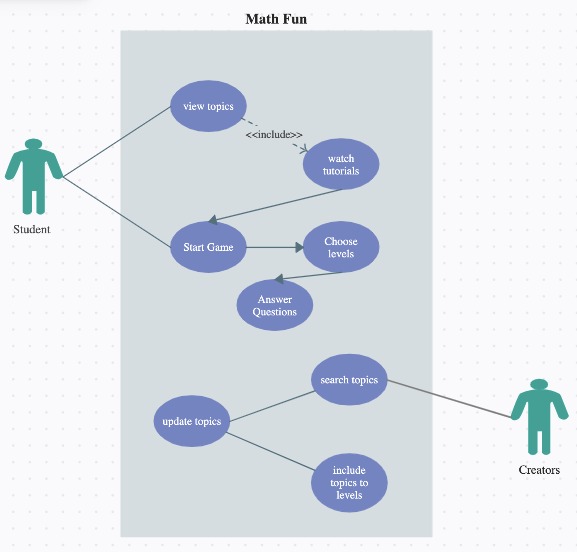


Figure 3. Use case diagram.

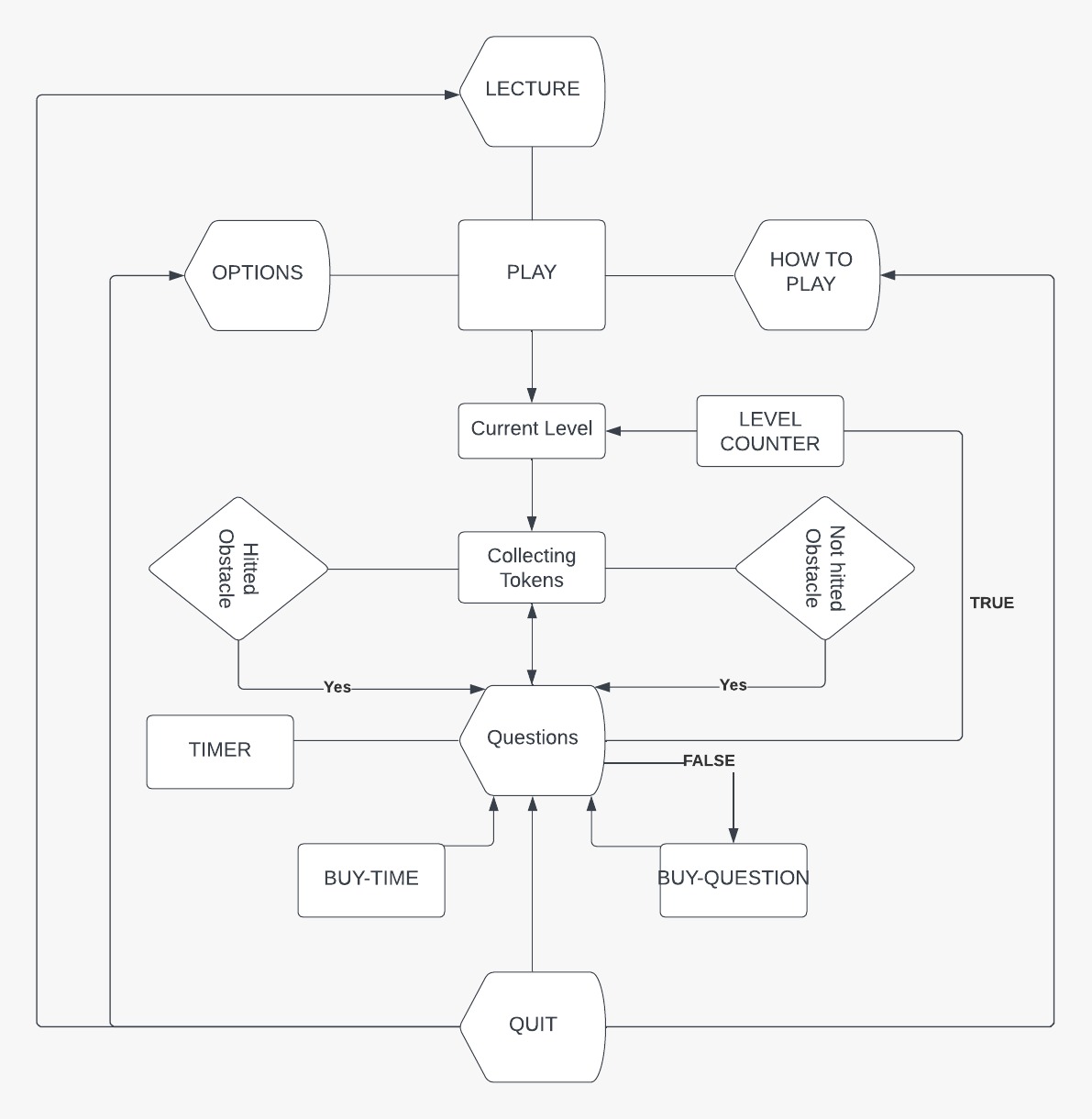


Figure 4. ER diagram.

Ch5: Design

* 1. Usability of the Design

The game’s main purpose is that pass the mathematic learning to children with the support of mobile game. Hence the target audience is the young part of the society; especially children between 14-17 ages, it decided to teach the decided math topic via some mobile app that supports the learning. When creating the mobile app, the Unity platform used, and it is decided to use android platform for spreading to the audience. When the platforms considered, it can clearly say that design is usable enough to publish and the spread the awareness of mathematical learning with mobile games.

* 1. Diagram Design

### Class Diagram of the System

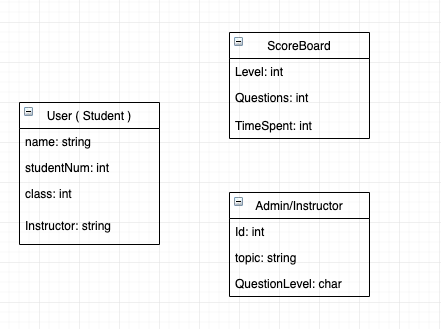


Figure 5. Class diagram.

Hence the project mainly is about a mobile game, there is not an exact class diagram. Nevertheless, it can be said that the figure above can be a class diagram when the actors considered. User, who is mainly the student, has a name in the type of string, studentNum in the type of int, class also an int and an instructor which is a string type. User should enter this information before joining the game. The scoreboard has level in the type of int, questions also an int which determines the number of the questions and TimeSpent also an int which determines the users time spent during the game. It should be mentioned that these values determined by system. Lastly the admin, which is the instructor of the class has Id, which is int type, chosen topic in the type of string and QuestionLevel in the type of char. The values determined by Instructor and passed to the admin to enhance the game.

### Use Case Diagram

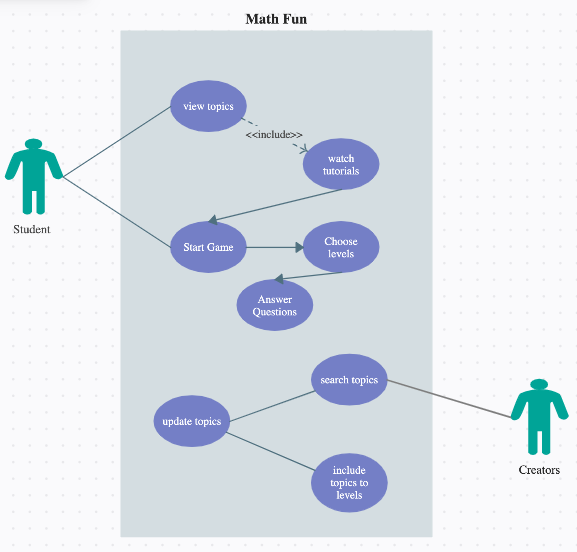
****

Figure 6. Use case diagram.

The figure above shows the use case diagram of the project. The project has mainly two different actors which is student and the creators. Student can view the topics, watch tutorials about these topics. And after that, he or she starts the game and then choose levels. After choosing the level, the game starts, and the student should answer questions truly for passing the levels. On the other hand, creators search current topics about permutations and combinations, update the topics due to current curriculum and include these topics to the levels in the order of question difficulties.

### User Interface Design

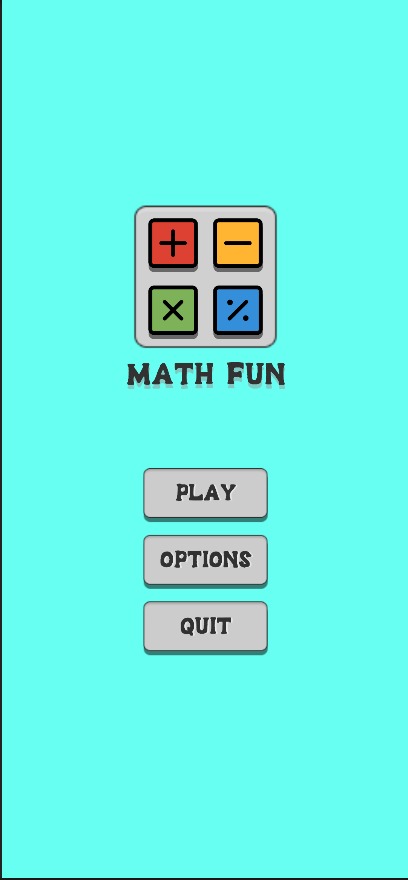


Figure 7. Start page.

In the start page there are three different buttons as shown in the figure above. The first button which is “Play” starts the game and let user play with the chosen level. Second button “Option” canalizes user to the options page where user can change the loudness of sounds of game. Last button “Quit” button closes the game.



Figure 8. First scene.

After pushing to the start button, this first scene will show up as the figure shown above. User can press play and start to play the game or can go to options or quit the game. Also, user can get some hints about the questions of the current level by pressing the hints button. There will be shown some hints about the chosen topic.

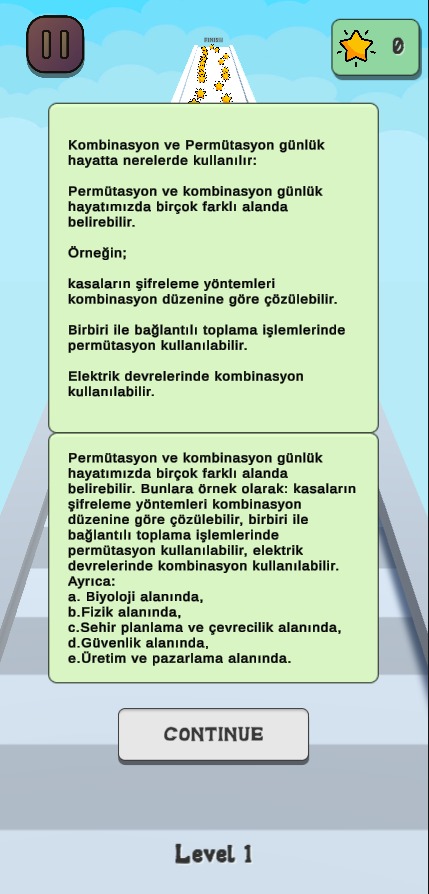


Figure 9. First tutorial.

In this part of the game; as shown in the figure above, system will give some short topic expression about the current level and let the user take some hints about the current topic. User should read all the tutorial to take some hints about the questions that he or she about to faced. After pressing continue button, user can start to play the game.

****

Figure 10. Question part.

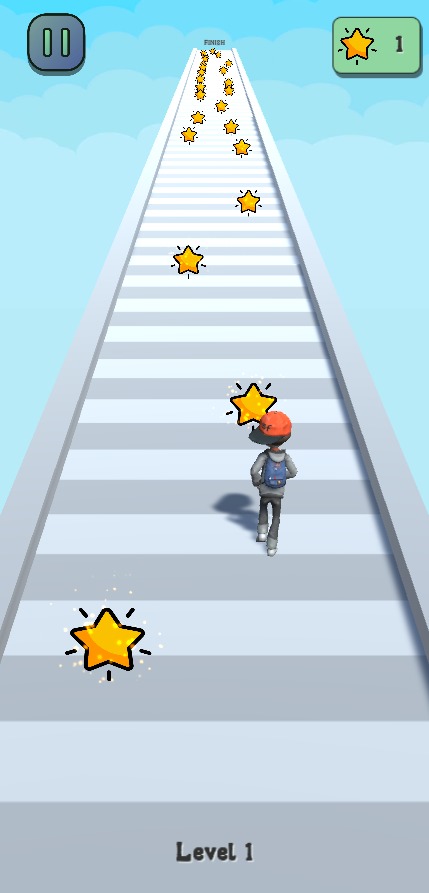
****

Figure 11. Collecting stars.

In the figure 10 above, the illustrations of the questions have given. Users face these questions during the game and should answer them truly to continue. In the figure 11, it has shown that user should collect starts in order to buy some time during answering questions or can take another question if he or she think that cannot answer the questions properly. Therefore, it can be said that the main purpose of the stars is buying time or taking new question.

### Software Development Methodology

In the MathFun project, the Spiral model has been used. As shown in the figure below 12, the spiral model has 4 different stages. It can be said that the Spiral model is the most efficient way to use when developing a mobile platform generated apps especially games. Hence it can be returned to the intended stage if there is a problem or bug during project, it has got massive usability for mobile platform generated games. Because it is known that there can be many errors in mobile apps before, during or after the publishing. In the first stage of spiral model, we determine objectives and identify alternative solutions due to the intended problem or project. In second stage, the project team identifies and resolves the opposing risks. In third stage of the design, the developing stage of the last version of the product has begun. Lastly in the fourth stage, general review has been done and the plan for the next stage will be started.

****

Figure 12. Spiral model.

Firstly, our project team decided the main objective as a learning of permutation and combination for the high school students. After the decision of the topic, the team identified and resolved the risks that may occur during and after the development stage. Later, our project team developed the final version of the product. Finally, the project team has reviewed the last version of the product and started to plan the next phases such as publishing the product under the supervision of our instructor.

### Timeline

In the figure 13 below, the timeline of the whole project has been given. Most important dates such as deadlines, reviews, code reviews, feedbacks et cetera has given as a milestone. Every team member has decided the dates of their own duties and did their duties according to these dates. There were not any unexpected, passed deadlines due to the milestones.

****

Figure 13. Timeline table.

### Budget and Cost

As every other project, MathFun has a certain cost. There were several costs that covered by the group members such as transportation and vital costs, electricity costs, network costs, publishing costs, poster and presentation costs and equipment costs. The transportation and vital costs such as beverages and meals were nearly total of 1000 Turkish liras. We used one of our team members studios to develop our project and the electricity and network cost were 200 Turkish Liras per month (3 months). The poster and presentation costs were 300 Turkish Liras due to some paperwork.

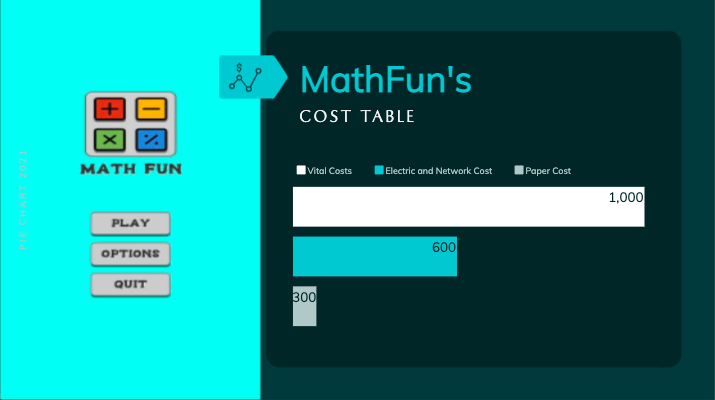
****

Figure 14. Cost table.

The values in the table above, has in Turkish Liras. Vital costs were 1000 Turkish liras, Electric and Network costs were 600 Turkish liras and lastly, the paper costs that has been used in presentation and poster were 300 Turkish liras.

### Collaboration

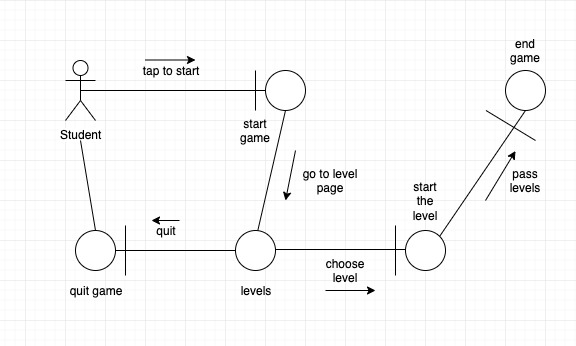


Figure 15. Collaboration diagram.

Ch6: Implementation & Test

In this project, we created a game that is beneficial for high school students who are struggling to learn permutation and combination subjects only getting education from the school. Meanwhile creating this game C# was used.

* 1. Implementation

C# software language and Unity were used to design all the features that a runner game should have and to enable students to fully use the actions such as movement, navigation, action, and also in-game features.

Unity was used to design this game. In Unity, the codes mentioned for both hardcoded and UI were used, and the implementation of the game was provided. The models, animations, and icons used in the game, it was integrated into Unity using resources from auxiliary websites such as flaticon and mixamo. The game's codes are reserved on GitHub and the copyrights of the game's designers are protected.

We created the button variable and accessed the component of the Pause and Resume buttons.



Figure 16. Pause & resume button.

When we press the Pause button, we activate the Pause panel variable in the GameUI script. We turn off the icon and buttons. We pause the game by making the game timescale zero.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 17. Pause function.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 18. Game stop screen.

When we press the continue button in the active pause panel variable, the game continues. The timescale becomes 1.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 19. Resume function.

When the Pause and Resume buttons are clicked, the functions shown in Figure 19 work.



Figure 20. Pause & resume button onClick.

In the Play button of the game, we created the button variable in the same way and accessed the component.



Figure 21. Play button.

When the Play button is pressed, the value of all the buttons appearing on the screen becomes false, disappears from the screen and the game starts. Also, the coroutine function is activated, it will be mentioned in Figure 21.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 22. Play function.

metin, gök, işaret içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 23. Start screen.

metin, gök, resim çerçevesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 24. Gameplay.

The function works when the Play button is clicked.



Figure 25. Play button onClick.

When we look at the coroutine function, when we press the Play button, the coroutine function runs within the Play function. That is, it is called from the onClick component inside the button.



Figure 26. First hint onClick.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 27. Coroutine function call.

At the beginning of the game, information about permutations and combinations appears. After the game starts, the information screen comes up after 1.25 seconds. SetActive takes the value true, and the screen opens. Using the DOTWEEN library, the information screen is displayed as a pop-up animation.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 28. Coroutine function.

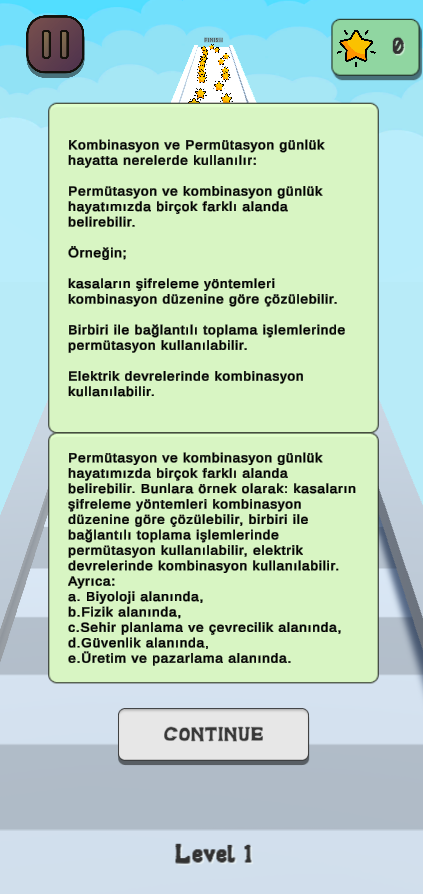


Figure 29. First hint screen.

When we click the Continue button, SetActive takes the value false, and the information screen disappears. Thus, the game continues.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 30. Continue button function.



Figure 31. Continue button.

When the user wants to get information about the subject before starting the game, he/she presses the subject narration part. Here, too, by creating a button, you can access the lecture section.



Figure 32. Hint button.

Buttons and icons disappear from the screen, SetActive takes the value true and the lecture appears.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 33. Hint function.

When the main menu button is pressed, the game returns to the main menu, the narration takes the value false, the buttons and icons appear on the screen with the value true.



Figure 34. Back to menu option.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 35. Back to menu function.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 36. Hint screen.

There is more than one screen in the subject narration section. When switching between these screens.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 37. The function of hint’s movements.

When the right arrow button is pressed on the screen, the page advances by taking the value +1. When you press the left arrow button, it takes the value of -1 and returns to the previous page.

When the mentioned buttons are clicked, the functions in Figure 37 work.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 38. Hints onClick.

When we examine the character movements;

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 39. Character movements.

Using these variables, we determine the speed of the character, the speed of going right and left, the capacity to move right and left. In addition, the animations of the character are also included in this variable.



Figure 40. Animation.

When our game starts, our character's run animation takes the value true, and the move function is called.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 41. Character movements function.

When our movement function is called, the character's movements are managed within this function. Since it is a runner game, the character moves forward automatically. But for the character to go right and left, the GetMouseButton(0) value must be kept pressed on the left mouse button and the character speed must be greater than zero. Depending on the mouse position, the character moves to the right or left within the swerveRange.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 42. Handle movement function.

gök, ışık, vektör grafikler içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 43. Gameplay.

metin, gök, ışık, demiryolu içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 44. Gameplay.

While collecting the character's stars during the game, Unity's own function is used. The character collider is the value of the collider of the object whose tag name is collectable. The score increases by 1 for each star we collect. The score animation plays as a pop-up and the score text is updated. The object we collect disappears. If the character collider touches the collider of the object whose tag name is QuestionCollider, a question arises. The Mesh option of the questions is kept off, only the collider value is used. When the character completes the level, the Confetti effect is played and the level completion panel opens with a value of true.In addition, the animation of our character is stopped with the value false.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 45. Colliders function.

metin, gök, ekran görüntüsü, vektör grafikler içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 46. Gameplay.



Figure 47. Question screen.



Figure 48. Level complete screen.

Collected stars are found with the tag named Collectables. The trigger value is on for the star to disappear after the character passes over the star.

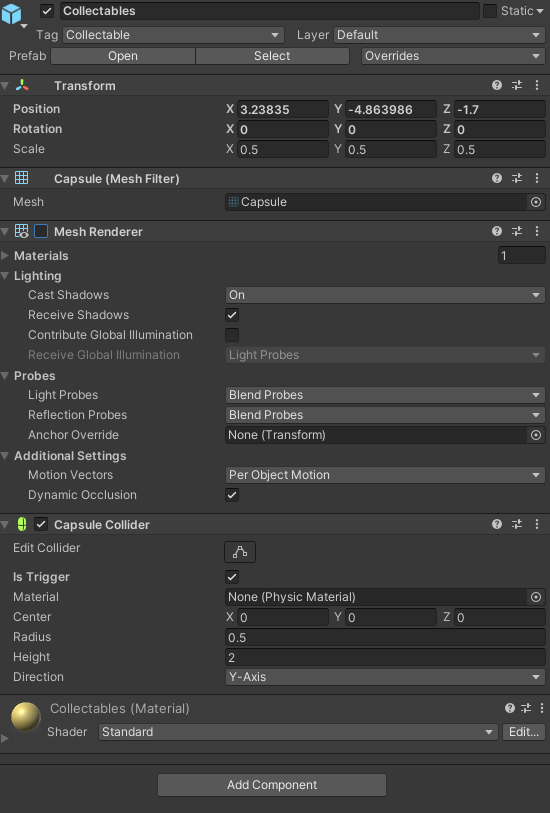
. 

Figure 49. Stars’ inspector.

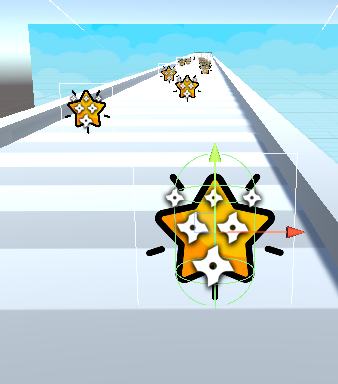


Figure 50. Stars’ image.

The questions we come across are found with the tag named Question Collider. Mesh value is turned off so that it does not appear on the screen after starting the game. Likewise, after the user answers the question, the question disappears from the screen.

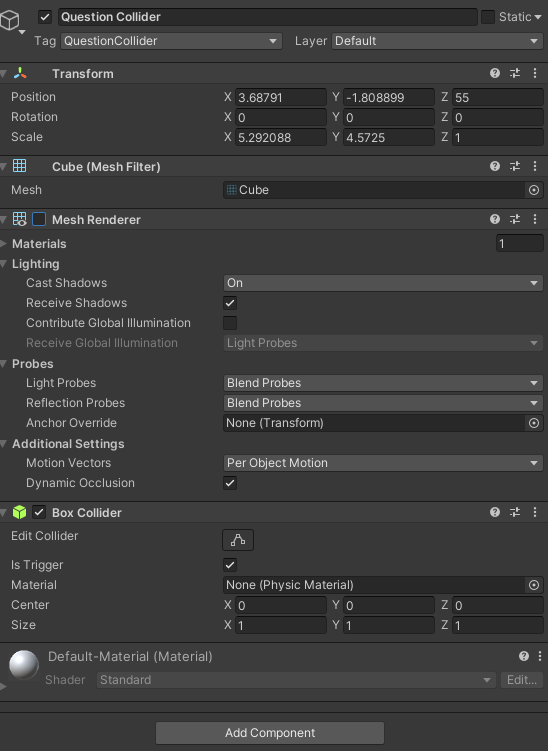


Figure 51. Questions’ inspector.

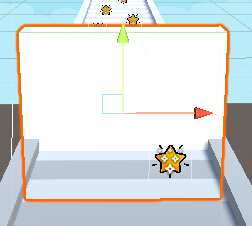


Figure 52. Question image.

6.2. Test

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST CASE** | **FUNCTION** | **ROLE** | **EXPECTED RESULT** |
| **CASE 1** | Start Game | Student(Default) | Success |
| **CASE 2** | Display “Konu Anlatımı” | Student(Default) | Success |
| **CASE 3** | Display “Nasıl Oynanır” | Student(Default) | Success |
| **CASE 4** | Display “Seçenekler” | Student(Default) | Success |
| **CASE 5** | Display “Çıkış” | Student(Default) | Success |
| **CASE 6** | Display “Oyna” | Student(Default) | Success |
| **CASE 7** | Click to “Oyna” | Student(Default) | Success |
| **CASE 8** | Click to “Konu Anlatımı” | Student(Default) | Success |
| **CASE 9** | Click to “Nasıl Oynanır” | Student(Default) | Success |
| **CASE 10** | Click to “Seçenekler” | Student(Default) | Success |
| **CASE 16** | Click to “Çıkış” | Student(Default) | Success |
| **CASE 17** | Collecting Tokens | Student(Default) | Success |
| **CASE 1** | Start Game | Student(Default) | Success |
| **CASE 11** | Dodging Obstacle | Student(Default) | Success |
| **CASE 12** | Hitting Obstacle | Student(Default) | Success |
| **CASE 13** | Level Achievement | Student(Default) | Success |
| **CASE 14** | Buying Time | Student(Default) | Success |
| **CASE 15** | Buying New Question | Student(Default) | Success |
| **CASE 16** | Click to Answer | Student(Default) | Success |
| **CASE 17** | Displaying True Solution | Student(Default) | Success |
| **CASE 9** | Displaying Timer | Student(Default) | Success |
| **CASE 10** | Display Answer Options | Student(Default) | Success |
| **CASE 16** | Wrong Answer Alert | Student(Default) | Success |
| **CASE 17** | True Answer Alert | Student(Default) | Success |

Ch7: Evaluation

“MathFun” is unity developed mobile game that supports mathematic learning and, it is planned that the game will be online on both in App Store and Google Play Store. MathFun allows students to learn permutation and combination topic with the support of a runner game. MathFun enables teachers to track the students’ achievements though all levels and questions. Students can download app to their smartphones and log in with their student id’s. Teachers act as admin in the flow and can communicate with the actual admins which is developer. Teachers can demand any upgrade through all levels and can change any questions according to curriculum. By courtesy of this feature, it can be said that MathFun will be always up to date. Currently, game has 5 different levels and in every level students have to face with different questions that pops randomly. Students have to answer all questions correctly to pass levels. The applications’ main purpose is to support mathematic learning with some fun. Generally, it can be said that the application will keep its development through current curriculum and teachers demand.

Ch8: Impact of the Project & Compliance with the Constraints

8.1. Compliance with the Realistic Constraints

Table 2. Realistic constraints & condition.

|  |  |
| --- | --- |
| Economic Factors | Please specify/explain realistic constraints and conditions (type, use, amount, etc.) |
|  | |
| EXPENDITURES |  |
| Computer | Used project members computers |
| Other Devices | NA |
| Peripherals | NA |
| Internet Connection | University’s connection and members own connection |
| Software | Unity and Visual Studio Code |
| Textbook/Magazine/Support Material | Mixamo , Flaticon |
| Human Resources | There were 3 members in the project team |
| Other | NA |
|  | |
| FUNDING Sources |  |
| University Resources | University’s connection and library |
| Project support (SANTEZ, TÜBİTAK, and so on) | TÜBİTAK (the idea is from there) |
| Support by the Industry | NA |
| Self-funded | Members used own sources such as computer and network |
| Other sources | Open Source Frameworks |
|  | |
| OTHER CONSTRAINTS |  |
| Memory | Memory will increase constantly due to number of users. |
| Runtime Efficiency | It depends on smartphone’s CPU |
| MANAGERIAL |  |
| Schedule (time) | The whole project took one semester (3 to 4 months) to develop. |

8.2. Impact of the Project

As whole project team the project has been evaluated. The positive and the negative aspects of the project has been decided. It is seen that the idea of uniting mathematics with mobile game has positive impact among students. The expected result was that support mathematic learning with the help of games and increase the ease of learning. It is seen that the idea of the learning with games is promising.

Table 3. Impact assessment report.

|  |  |
| --- | --- |
| Professional/Ethical Issues | Please specify/explain (existence of items, violation of items, awareness about items) |
|  | |
| ETHICS/IT Law |  |
| Copyright in copying multimedia (sound, video, text) | The animations of the project are taken from open source platforms. |
| Use of licensed software | Github, Unity Hub |
| Data Privacy | User names, ids and the passwords are end to end encrypted |
| Use of patented products/ideas | TÜBİTAK’s project idea |
| IT Laws in Turkey (5661 and others) | NA |
| IT Laws - International | NA |
| Plagiarism | All of the codes are self-written |
|  | |
| PROFESSIONAL |  |
| Sustainability (use of Licensed and/or open source code) | Licensed softwares such as unity |
| Maintenance | Maintenance of the app is done throughout feedbacks and bugs. |
| Liability | Admins are responsible for storing the passwords of the users. |
| Financial impact/Manufacturability | Used sources were members. Therefore, financing were affordable and reasonable. |
|  |  |
| SOCIAL/POLITICAL/  ENVIRONMENTAL |  |
| Political impact | NA |
| Impact on health | NA |
| Gambling | NA |
| Pornography | NA |
| Equal Access/equity | All of the students and the teachers can access easily. |
| Environmental impacts (energy, carbon footprint and so on) | NA |
| Technology acceptance & Human/Business psychology | The system was developed as a learning game. |
| Security issues | The passwords of the students can be accessed by teachers and it could be an issue. |
|  |  |
| PROFESSIONAL (CODES, STANDARDS, FRAMEWORKS) |  |
| IEEE | NA |
| ISO | NA |
| ANSI | NA |
| TSE | NA |
| ITIL/COBIT | NA |
| OTHER | NA |

Ch9: Conclusion

During the development process of this project, we got an idea about the difficulties we may encounter while taking part in a project in business life. We have experienced the problems that can arise while developing a project or application and how we can fix these problems. During the project development process, we had the opportunity to apply the training of many theoretical and applied courses we took in our courses in our faculty, and we learned new techniques. We consolidated a lot of information we know and learn. We took care to work as a harmonious team and took care not to have conflicts within the team. We shared work according to our abilities and ensured that everyone in the team contributed to every stage of the project. We helped each other at the points we did not know and could not do. We had an idea about the responsibilities we will take in business life and the risks we may face.

We had the opportunity to improve our coding skills during the software development process of our project. We did the necessary research at the stages we didn't know about and corrected our deficiencies. We named our game "MathFun". We used Unity and its integrated Visual Studio Code on the interface and coding side of the game. We used C# as the programming language. We got the figures and boxes in the game ready from the internet.

In conclusion, the development purpose of our project was to facilitate the learning of high school students about permutation-combination in the mathematics course. Thanks to our game, students will learn the topic we have chosen in a simple way with fun. In this way, they will understand that even the math lesson can be learned easily. In the future, we plan to publicize our project by improving it and adding new features. We think that many high schools need such an application. Thus, we think that the number of users will increase. With the advancement of technology, our app or similar apps will be used to facilitate learning in the education system.

References

Bütüner, S., Güler, M. (2017). "Gerçeklerle Yüzleşme: Türkiye’nin TIMSS Matematik Başarısı Üzerine Bir Çalışma," *BAYBURT EĞİTİM FAKÜLTESİ DERGİSİ*, vol.12, pp.353-376. Retrieved from <https://dergipark.org.tr/tr/download/article-file/317332>

Chizary, F., & Farhangi, A. (2017). Efficiency of Educational Games on Mathematics Learning of Students at Second Grade of Primary School. Journal of History Culture and Art Research, 6, 232-240.

Gün, H. K., Işık, O. R. & Şahin, B. (2021). OYUNLA ÖĞRETİMİN OLASILIK BAŞARISINA VE MATEMATİK DERSİNE TUTUMA ETKİSİ. Mustafa Kemal Üniversitesi Eğitim Fakültesi Dergisi, 5 (7), 263-276. Retrieved from <https://dergipark.org.tr/tr/pub/mkuefder/issue/63331/842558>

Kebritchi, M., Hirumi, A., Bai, H. (2010). The Effects Of Modern Mathematics Computer Games On Mathematics Achievement And Class Motivation, Computers & Education, 55 (2), 427-443, ISSN 0360-1315, https://doi.org/10.1016/j.compedu.2010.02.007.

Lopez-Morteo, G., López, G. (2007). Computer Support for Learning Mathematics: A Learning Environment Based on Recreational Learning Objects, Computers & Education, 48 (4), 618-641, ISSN 0360-1315, <https://doi.org/10.1016/j.compedu.2005.04.014>.

Mercan, M., Filiz, A., Göçer, İ., & Özsoy, N. (2009). Bilgisayar Destekli Eğitim Ve Bilgisayar Destekli Öğretimin Dünyada Ve Türkiye’de Uygulamaları. Şanlıurfa: Akademik Bilişim. Retrieved from <https://ab.org.tr/ab09/kitap/mercan_filiz_AB09.pdf>

Raja, R., & Nagasubramani, P. C. (2018). Impact of modern technology in education. Journal of Applied and Advanced Research, 3(1), 33–35. https://doi.org/10.21839/jaar.2018.v3is1.165

Rhode Island Department of Education. (n.d.). Curriculum Definition. Retrieved from <https://www.ride.ri.gov/InstructionAssessment/Curriculum/CurriculumDefinition.aspx>

Topçu, H., Küçük, S. & Göktaş, Y. (2014). Sınıf Öğretmeni Adaylarının İlköğretim Matematik Öğretiminde Eğitsel Bilgisayar Oyunlarının Kullanımına Yönelik Görüşleri. Turkish Journal of Computer and Mathematics Education (TURCOMAT) , 5 (2) , 119-136 . Retrieved from <https://dergipark.org.tr/tr/pub/turkbilmat/issue/21573/231476>

Appendix

Game UI:

using DG.Tweening;

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

using UnityEngine.UI;

public class GameUI : MonoBehaviour

{

[Header("SCRIPTS")]

public PlayerController player;

[Header("GAMEOBJECTS")]

public GameObject buttons;

public GameObject icon;

public GameObject optionsPanel;

public GameObject pausePanel;

public GameObject levelCompletePanel;

public GameObject hintsPanel;

public GameObject firstHintPanel;

public GameObject howToPlayPanel;

public GameObject levelTextObject;

[Header("ARRAY")]

public GameObject[] hintsImg;

[Header("BUTTONS")]

public Button play;

public Button options;

public Button quit;

public Button pause;

public Button resume;

public Button hints;

public Button nextHint;

public Button backHint;

public Button backToMenu;

public Button continueHint;

public Button howToPlay;

public Button back;

public Button nextLevel;

[Header("ANIMATIONS")]

public Animation buttonPopUpAnim;

[Header("PARTICLE")]

public ParticleSystem confetti;

[Header("BOOL")]

public bool isGameStart;

public bool isGameFinish;

public bool isLevelComplete;

public bool isLevelFail;

[Header("INT")]

public int hintIndex;

void Start()

{

player = FindObjectOfType<PlayerController>();

hintIndex = 0;

Button playButton = play.GetComponent<Button>();

Button optionsButton = options.GetComponent<Button>();

Button quitButton = quit.GetComponent<Button>();

Button pauseButton = pause.GetComponent<Button>();

Button resumeButton = resume.GetComponent<Button>();

Button hintsButton = hints.GetComponent<Button>();

Button backToMenuButton = backToMenu.GetComponent<Button>();

Button nextHintButton = nextHint.GetComponent<Button>();

Button backHintButton = backHint.GetComponent<Button>();

Button firstHintContinueButton = continueHint.GetComponent<Button>();

Button howToPlayButton = howToPlay.GetComponent<Button>();

Button backButton = back.GetComponent<Button>();

Button nextLevelButton = nextLevel.GetComponent<Button>();

playButton.onClick.AddListener(PlayOnClick);

optionsButton.onClick.AddListener(OptionsOnClick);

quitButton.onClick.AddListener(QuitOnClick);

pauseButton.onClick.AddListener(PauseOnClick);

resumeButton.onClick.AddListener(ResumeOnClick);

hintsButton.onClick.AddListener(HintsOnClick);

backToMenuButton.onClick.AddListener(BackToMenuOnClick);

nextHintButton.onClick.AddListener(NextHintOnClick);

backHintButton.onClick.AddListener(BackHintOnClick);

firstHintContinueButton.onClick.AddListener(FirstHintContinue);

howToPlayButton.onClick.AddListener(HowToPlayOnClick);

backButton.onClick.AddListener(Back);

nextLevelButton.onClick.AddListener(LoadLevel);

}

void Update()

{

}

void PlayOnClick()

{

play.GetComponent<Animation>().Play();

isGameStart = true;

icon.SetActive(false);

buttons.SetActive(false);

optionsPanel.SetActive(false);

levelTextObject.SetActive(false);

StartCoroutine(FirstHintShowCorotine());

}

void OptionsOnClick()

{

buttons.SetActive(false);

options.GetComponent<Animation>().Play();

optionsPanel.SetActive(true);

icon.SetActive(false);

levelTextObject.SetActive(false);

}

void QuitOnClick()

{

quit.GetComponent<Animation>().Play();

}

void PauseOnClick()

{

if (!isGameStart)

{

icon.SetActive(false);

buttons.SetActive(false);

levelTextObject.SetActive(false);

}

optionsPanel.SetActive(false);

hintsPanel.SetActive(false);

pausePanel.SetActive(true);

Time.timeScale = 0;

}

void ResumeOnClick()

{

if (!isGameStart)

{

icon.SetActive(true);

buttons.SetActive(true);

levelTextObject.SetActive(true);

}

pausePanel.SetActive(false);

Time.timeScale = 1;

}

void HintsOnClick()

{

icon.SetActive(false);

buttons.SetActive(false);

optionsPanel.SetActive(false);

hintsPanel.SetActive(true);

levelTextObject.SetActive(false);

}

public void BackToMenuOnClick()

{

hintsPanel.SetActive(false);

icon.SetActive(true);

buttons.SetActive(true);

optionsPanel.SetActive(false);

levelTextObject.SetActive(true);

}

void NextHintOnClick()

{

if (hintsImg[hintIndex+1] != null)

{

hintsImg[hintIndex].SetActive(false);

hintsImg[hintIndex + 1].SetActive(true);

hintIndex += 1;

}

}

void BackHintOnClick()

{

if (hintsImg[hintIndex-1] != null)

{

hintsImg[hintIndex].SetActive(false);

hintsImg[hintIndex - 1].SetActive(true);

hintIndex -= 1;

}

}

public IEnumerator FirstHintShowCorotine()

{

yield return new WaitForSeconds(1.25f);

firstHintPanel.SetActive(true);

firstHintPanel.transform.DOScale(new Vector3(1.5f,1.5f,1.5f),0.25f).OnComplete(()=>

firstHintPanel.transform.DOScale(Vector3.one,0.25f));

//Time.timeScale = 0;

}

void FirstHintShow()

{

StartCoroutine(FirstHintShowCorotine());

}

void FirstHintContinue()

{

firstHintPanel.SetActive(false);

//Time.timeScale = 1;

}

void HowToPlayOnClick()

{

buttons.SetActive(false);

icon.SetActive(false);

howToPlayPanel.SetActive(true);

}

void Back()

{

howToPlayPanel.SetActive(false);

buttons.SetActive(true);

icon.SetActive(true);

}

public void LoadLevel()

{

// load the nextlevel

SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

}

}

Lock Camera X:

using UnityEngine;

using Cinemachine;

/// <summary>

/// An add-on module for Cinemachine Virtual Camera that locks the camera's Z co-ordinate

/// </summary>

[ExecuteInEditMode]

[SaveDuringPlay]

[AddComponentMenu("")] // Hide in menu

public class LockCameraZ : CinemachineExtension

{

[Tooltip("Lock the camera's Z position to this value")]

public float m\_XPosition = 0;

protected override void PostPipelineStageCallback(

CinemachineVirtualCameraBase vcam,

CinemachineCore.Stage stage, ref CameraState state, float deltaTime)

{

if (stage == CinemachineCore.Stage.Body)

{

var pos = state.RawPosition;

pos.x = m\_XPosition;

state.RawPosition = pos;

}

}

}

Movement Handler:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MovementHandler : MonoBehaviour

{

public Transform player;

public float speed;

public float swerveRange;

public float swerveSpeed;

public float rotationSpeed;

public Animator anim;

public GameUI gameUI;

void Start()

{

gameUI = FindObjectOfType<GameUI>();

}

void Update()

{

if (Input.GetMouseButtonDown(0))

{

lastPos = Input.mousePosition.x;

}

HandleMovement();

}

float? lastPos = null;

void HandleMovement()

{

float ratio = Screen.width / (swerveRange \* 2 \* swerveSpeed);

transform.Translate(Vector3.forward \* speed \* Time.deltaTime, Space.World);

if (Input.GetMouseButton(0))

{

var touchPoint = Input.mousePosition.x - lastPos.GetValueOrDefault(Screen.width / 2);

lastPos = Input.mousePosition.x;

Vector3 startPos = player.position;

Vector3 targetPos = new Vector3(Mathf.Clamp(startPos.x + touchPoint / ratio, -swerveRange, swerveRange), player.position.y, player.position.z);

player.position = targetPos;

var rot = Quaternion.Euler(0, Mathf.Clamp(touchPoint \* 100, -45, 45), 0);

player.rotation = Quaternion.Slerp(player.rotation, rot, rotationSpeed);

}

else

{

var rot = Quaternion.Euler(0, 0, 0);

player.rotation = Quaternion.Slerp(player.rotation, rot, rotationSpeed);

}

if (!anim.GetBool("isRunning"))

anim.SetBool("isRunning", true);

}

}

Obstacle Controller:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using DG.Tweening;

public class ObstacleController : MonoBehaviour

{

[Header("FLOAT")]

public float speed;

void Start()

{

transform.DOMoveZ(-20,10).OnComplete(()=>

Destroy(gameObject));

}

void Update()

{

}

}

Obstacle Spawner:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ObstacleSpawner : MonoBehaviour

{

[Header("SCRIPTS")]

public GameUI gameUI;

[Header("GAMEOBJECT")]

public GameObject obstaclePrefab;

public GameObject question;

public GameObject[] obstacles;

[Header("FLOAT")]

public float time;

public float repeatRate;

[Header("INT")]

public int spawnAmount;

[Header("BOOL")]

public bool isSpawnAvaible;

void Start()

{

spawnAmount = 0;

gameUI = FindObjectOfType<GameUI>();

InvokeRepeating("SpawnObstacle", time, repeatRate);

}

void Update()

{

if (gameUI.isGameStart)

{

question = GameObject.FindGameObjectWithTag("Question");

obstacles = GameObject.FindGameObjectsWithTag("Obstacle");

if (question != null)

{

spawnAmount = 0;

for (int i = 0; i < obstacles.Length; i++)

{

Destroy(obstacles[i]);

}

}

else

{

spawnAmount = 1;

}

}

}

public void SpawnObstacle()

{

for (int i = 0; i < spawnAmount; i++)

{

GameObject obstacle = Instantiate(obstaclePrefab, new Vector3(Random.Range(-1f, 2f), 1.5f, transform.position.z), Quaternion.Euler(0, 180, 0));

}

}

}

Player Controller:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class PlayerController : MonoBehaviour

{

[Header("SCRIPTS")]

public GameUI gameUI;

[Header("FLOAT")]

public float speed;

public float swerveSpeed;

public float swerveRange;

[Header("INT")]

public int score;

public int questionIndex;

public int newQuestionsIndex;

[Header("BOOL")]

public bool isCountDownStart;

public bool isQuestionSpawned;

public bool isStopped;

[Header("GAMEOBJECT")]

public GameObject beforeTimer;

public GameObject timer;

public GameObject timesUp;

public GameObject question;

public GameObject spawnedQuestion;

[Header("ARRAY")]

public GameObject[] questionPrefabs;

public GameObject[] newQuestions;

[Header("ANIMATOR")]

public Animator anim;

[Header("ANIMATION")]

public Animation scoreTextPopUpAnim;

[Header("TRANSFORM")]

public Transform player;

[Header("TEXT")]

public Text scoreText;

public Text timeText;

public Text beforeTimerText;

void Start()

{

gameUI = FindObjectOfType<GameUI>();

anim = GetComponent<Animator>();

//player = transform;

questionIndex = 0;

newQuestionsIndex = 0;

}

void Update()

{

if (gameUI.isGameStart && !gameUI.isGameFinish)

{

spawnedQuestion = GameObject.FindGameObjectWithTag("Question");

scoreText.text = score.ToString();

anim.SetBool("Run",true);

HandleMovement();

if (spawnedQuestion != null)

{

Stop();

}

else

{

Continue();

StopAllCoroutines();

}

}

if (Input.GetMouseButtonDown(0))

{

//lastPos = Input.mousePosition.x;

}

}

float? lastPos = null;

void HandleMovement()

{

float ratio = Screen.width / (swerveRange \* 2 \* swerveSpeed);

transform.Translate(Vector3.forward \* speed \* Time.deltaTime, Space.World);

if (Input.GetMouseButton(0) && speed > 0)

{

var touchPoint = Input.mousePosition.x - lastPos.GetValueOrDefault(Screen.width / 2);

lastPos = Input.mousePosition.x;

Vector3 startPos = player.position;

Vector3 targetPos = new Vector3(Mathf.Clamp(startPos.x + touchPoint / ratio, -swerveRange, swerveRange), player.position.y, player.position.z);

player.position = targetPos;

}

}

public float currCountdownValue;

public IEnumerator StartCountdown(float countdownValue = 60)

{

currCountdownValue = countdownValue;

while (currCountdownValue > 0)

{

timeText.text = currCountdownValue.ToString();

Debug.Log("Countdown: " + currCountdownValue);

yield return new WaitForSeconds(1.0f);

currCountdownValue--;

if (currCountdownValue==0)

{

timer.SetActive(false);

timesUp.SetActive(true);

yield return new WaitForSeconds(0.5f);

timesUp.SetActive(false);

Destroy(spawnedQuestion);

break;

}

}

}

float time;

public IEnumerator BeforeTimerStart(float countdownValue = 3)

{

yield return new WaitForSeconds(1f);

beforeTimer.SetActive(true);

time = countdownValue;

beforeTimerText.text = "Sürenin başlamasına " + time.ToString();

while (countdownValue>0)

{

beforeTimerText.text = "Sürenin başlamasına " + time.ToString();

yield return new WaitForSeconds(1.0f);

time--;

if (time == 0)

{

beforeTimer.SetActive(false);

timer.SetActive(true);

StartCoroutine(StartCountdown());

break;

}

}

}

public void BuyTime(int cost)

{

score -= cost;

scoreTextPopUpAnim.Play();

scoreText.text = score.ToString();

currCountdownValue += 20f;

}

public void BuyQuestion(int cost)

{

score -= cost;

scoreTextPopUpAnim.Play();

Destroy(spawnedQuestion);

Instantiate(newQuestions[newQuestionsIndex], new Vector3(-1.25f, 1.5f, transform.position.z + 5), Quaternion.identity);

newQuestionsIndex += 1;

}

public void Stop()

{

speed = 0;

anim.SetBool("Run", false);

}

public void Continue()

{

speed = 5;

anim.SetBool("Run", true);

}

private void OnCollisionEnter(Collision collision)

{

if (collision.gameObject.CompareTag("Obstacle"))

{

//anim.SetBool("isHit",true);

anim.SetTrigger("isHit");

}

}

private void OnTriggerEnter(Collider other)

{

if (other.gameObject.CompareTag("Collectable"))

{

score += 1;

scoreTextPopUpAnim.Play();

scoreText.text = score.ToString();

Destroy(other.gameObject);

}

if (other.gameObject.CompareTag("QuestionCollider"))

{

if (!isQuestionSpawned)

{

Instantiate(questionPrefabs[questionIndex], new Vector3(-1.25f, 1.5f, transform.position.z + 5), Quaternion.identity);

questionIndex += 1;

StartCoroutine(BeforeTimerStart());

isQuestionSpawned = true;

}

}

if (other.gameObject.CompareTag("Finish"))

{

gameUI.confetti.Play();

gameUI.isGameFinish = true;

gameUI.levelCompletePanel.SetActive(true);

speed = 0;

anim.SetBool("Run",false);

}

}

private void OnTriggerExit(Collider other)

{

if (other.gameObject.CompareTag("QuestionCollider"))

{

isQuestionSpawned = false;

}

}

}

Question Controller:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using TMPro;

public class QuestionController : MonoBehaviour

{

[Header("SCRIPT")]

public PlayerController playerController;

[Header("BUTTONS")]

public Button buyTime;

public Button buyNewQuestion;

public Button trueAnswer;

public Button wrongAnswer;

public Button wrongAnswer\_1;

public Button wrongAnswer\_2;

public Button continueBt;

[Header("GAMEOBJETS")]

public GameObject trueText;

public GameObject falseText;

public GameObject question;

public GameObject continueButtonObject;

[Header("ARRAY")]

public GameObject[] answerButtons;

[Header("INT")]

public int timeCost;

public int questionCost;

[Header("TEXT")]

public TextMeshProUGUI timeCostText;

public TextMeshProUGUI questionCostText;

void Start()

{

timeCostText.text = timeCost.ToString();

questionCostText.text = questionCost.ToString();

playerController = FindObjectOfType<PlayerController>();

Button buyTimeButton = buyTime.GetComponent<Button>();

buyTimeButton.onClick.AddListener(BuyMoreTime);

Button buyNewQuestionButton = buyNewQuestion.GetComponent<Button>();

buyNewQuestionButton.onClick.AddListener(BuyNewQuestion);

Button trueButton = trueAnswer.GetComponent<Button>();

trueButton.onClick.AddListener(True);

Button wrongButton = wrongAnswer.GetComponent<Button>();

wrongButton.onClick.AddListener(False);

Button wrongButton\_1 = wrongAnswer\_1.GetComponent<Button>();

wrongButton.onClick.AddListener(False);

Button wrongButton\_2 = wrongAnswer\_2.GetComponent<Button>();

wrongButton.onClick.AddListener(False);

Button continueButton = continueBt.GetComponent<Button>();

continueButton.onClick.AddListener(ContinueOnClick);

}

void Update()

{

timeCostText.text = timeCost.ToString();

questionCostText.text = questionCost.ToString();

if (playerController.score<timeCost)

{

buyTime.interactable = false;

}

if (playerController.score<questionCost)

{

buyNewQuestion.interactable = false;

}

if (playerController.beforeTimer.activeSelf==true)

{

trueAnswer.interactable = false;

wrongAnswer.interactable = false;

wrongAnswer\_1.interactable = false;

wrongAnswer\_2.interactable = false;

buyNewQuestion.interactable = false;

buyTime.interactable = false;

}

else

{

trueAnswer.interactable = true;

wrongAnswer.interactable = true;

wrongAnswer\_1.interactable = true;

wrongAnswer\_2.interactable = true;

buyNewQuestion.interactable = true;

buyTime.interactable = true;

}

}

public void True()

{

trueText.SetActive(true);

playerController.score += 1;

playerController.scoreText.text = playerController.score.ToString();

StartCoroutine(Destroy());

}

public void False()

{

buyNewQuestion.gameObject.SetActive(false);

buyTime.gameObject.SetActive(false);

playerController.timer.SetActive(false);

question.SetActive(false);

falseText.SetActive(true);

for (int i = 0; i < answerButtons.Length; i++)

{

answerButtons[i].SetActive(false);

}

continueButtonObject.SetActive(true);

}

public void BuyMoreTime()

{

if (playerController.score>=timeCost)

{

playerController.BuyTime(timeCost);

timeCost += 2;

timeCostText.text = timeCost.ToString();

}

}

public void BuyNewQuestion()

{

if (playerController.score>=questionCost)

{

playerController.BuyQuestion(questionCost);

questionCost += 2;

questionCostText.text = questionCost.ToString();

}

}

public void ContinueOnClick()

{

Destroy(gameObject);

}

IEnumerator Destroy()

{

playerController.timer.SetActive(false);

yield return new WaitForSeconds(1.5f);

gameObject.SetActive(false);

Destroy(gameObject,1);

}

}

**Link:** https://github.com/Emrehan325/MathFun