**DOKUZ EYLUL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CME1252 PROJECT BASED LEARNING – II**

**FINAL REPORT**

**PROJECT – II**

**GRAVITY GAME**

**by**

**Ali Özgür İNEP**

**Ege YILDIRIM**

**İmdat SÖNMEZ**

**Talha Mustafa ANTEP**

**Lecturers**

**Arş. Gör. Fatih DİCLE**

**Doç. Dr. Mehmet Hilal ÖZCANHAN**

**28.04.2023**

# CHAPTER ONE

PROGRESS DESCRIPTION

Purpose of the project is to create a new game like boulder dash with the basic java knowledge we learned from Algorithm and Programming lessons. In addition to java knowledge, we had to discover ENIGMA library also. Roughly, we made a game that the player should get higher scores as much as possible. According to some choices of the player, the game will be shaped in many ways. We started on 27.03.2023 and finished on 27.04.2023.

# CHAPTER TWO

TASK SUMMARY

## Completed Tasks

Ali Özgür İNEP: Firstly, in addition to our first progress, boulder fall was added into our code. Respect to some game rules boulders are falling. In default settings, the boulders are falling to the space(if left is space boulder falls left, if right is space boulder falls right). Secondly, some additional improvements such as login screen, settings, some player choices which change the game rules. Robots movement already had been setted, however; the interaction with boulders was just adjusted. Conclusively, all of the tasks were completed.

İmdat SÖNMEZ: There was no clear distribution of tasks in the project because everyone tried to be in every part of the project. Some parts of the project were carried out jointly and some parts individually. Created playground, walls, earth squares, numbers, boulders. Input queue class has been created. Over time different items have been added and modified in the input queue. Time class was created and integrated into the game and used in different parts of the game. Worked on the pushing sideways and falling of the boulders but this part wasn't finished on my own. These codes were also made by my other group mates and the most efficient one was integrated into the main code. slides prepared together with Ege.

Talha Mustafa ANTEP: A class named backpack was created, and it was designed to hold numbers that contribute to the score and other attributes in the game. The initialization class was also created to initialize the game field, and the player’s attributes were defined within this field. As part of the project's second phase, we focused on addressing the issue of boulder fall. Initially, we encountered some errors, but were able to resolve them by utilizing Özgür's code and conducting a through debugging process. Once the necessary adjustments were made, we were able to successfully address the issue and create a workable solution. Finally, we compiled all of our findings and presented them in a comprehensive final report.

Ege YILDIRIM: Queue class was opened.Methods were added in CircularQueue format.Methods were written for the implementation of InputQueue.The errors that appeared in the Boulder fall section were debugged.Helped to colour the screen.Made a highscore table where the scores will appear at the end of the game.Helped to prepare the slide.Prepared a video introducing the game.

## Incomplete Tasks: Reasons and Explanations

There are no incomplete tasks.

## Additional Improvements ~~to the Project~~

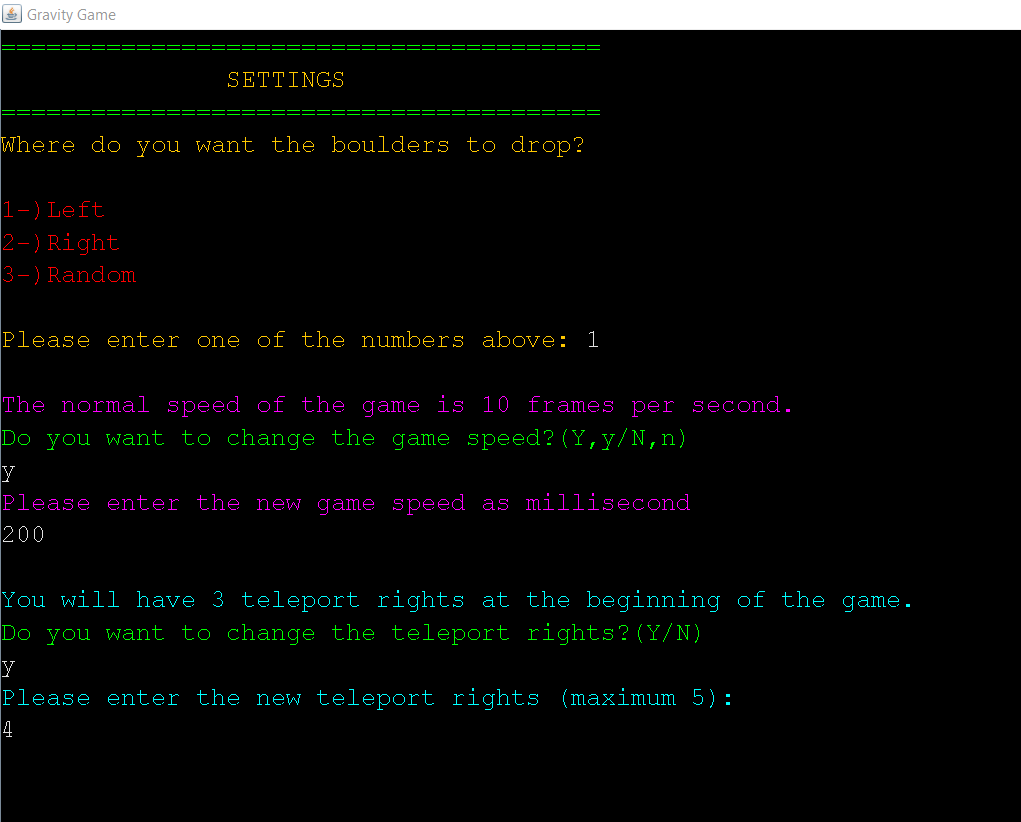
Coloring and enlargement were performed on the console. Some rules of the game which are able to be changed. Such as speed of the game, settings, login screen, highscore table.

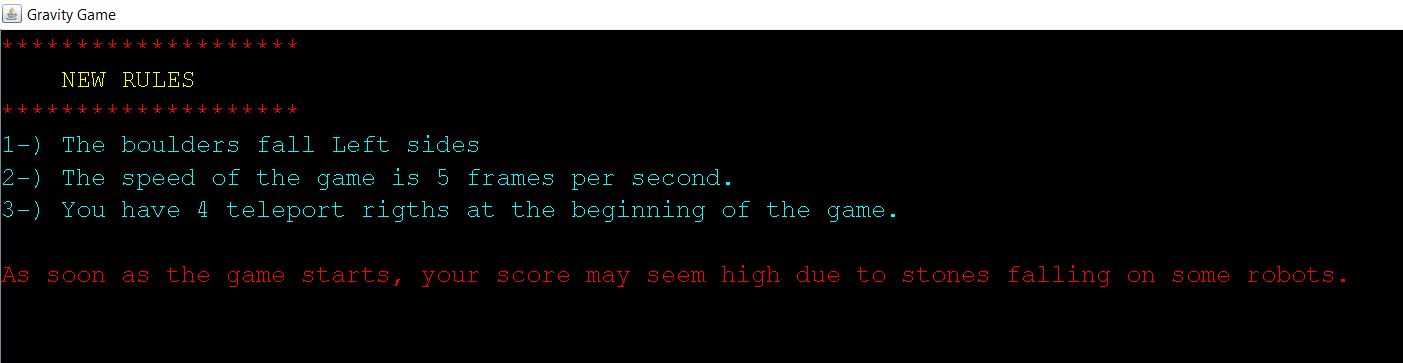
# CHAPTER THREE

EXPLANATION of algorıthms

## Screenshots

## 







## Functions

Almost, all of our Project consist of functions. Hence, there are some significant methods such as boulderFall(), placeBoulders(), and placeNumbers(). In addition to these, there are also display methods such as printBackpack(), gameArea(), and printTime().

## Algorithms and Solution Strategies

Ali Özgür İNEP: There are too many classes were created. Thus, dividing the Project into some pieces helps handling the operations. For instance, to handle the boulder fall, for-while loops, if-else and switch data structures were used. With for loop the boulders were found then, the below of the boulders were checked and the interactions were adjusted. We have completed the project with classes to highlight the "Object-oriented" logic.

İmdat SÖNMEZ: Firstly the playing field is kept in a two-dimensional array and then the game arena is filled with earth square. Then numbers, boulders, robots are placed. The algorithm for adding elements to the input queue is as follows: The probability of the elements to be added to the input queue is different from each other. So a random number is generated from 1-40. Each element has different intervals, for example the probability of 1 is 6/40. So if the generated random number comes between 1-6, 1 will be added to the input queue.

Talha Mustafa ANTEP: The game area was created using a two-dimensional array and certain requirements were met using for loops. Classes were used to minimize errors when setting up teleportation or robots, and setters, getters, and constructors were coded. Storing constants like player , walls or boulders helped with time control, and using Initialization. Overall, online sources were used to inform the coding process.

Ege YILDIRIM: Just before the start of the game loop, Queue was written to be random.Then, by calling the ElementsToArea command, the element was added to the map as one every 3 seconds.When creating the HighScoreTable, first all lines were read in the text file, then first the array and then added to the list.After the game was over, the name was taken from the user and new arrays were opened with the score and added to them.Names and scores were added to the lists according to the indexes of the arrays and printed in string format. The content of the text file was changed so that the name of the text file remained the same.

# CHAPTER FOUR

PROBLEMS ENCOUNTERED

Ali Özgür İNEP: There is no problem were faced in the code. Nevertheless, because of the earthquake in our country occurs some communication problems.

İmdat SÖNMEZ: Some simple errors occurred when integrating the code into the console screen, but easily solved. Other than that, no major problems occurred.

Talha Mustafa ANTEP: No big issues were encountered. Some problems were present with certain aspects of the stack and queue structures, but they were eventually resolved. I encountered most of the errors while working on the boulder fall. I received a null exception error while trying to locate and modify the positions of the boulders.

Ege YILDIRIM: While adding in InputQueue, there was a problem in adding a robot.It was adding the robot, but the robot was not moving and giving an outofbounds error.The problem was solved by writing the add method in the Robots class.There were deficiencies related to colouring.This problem was solved by asking other groups.Finally, it gave a null error when printing the list in highscoretable.The problem was solved by the team.

# CHAPTER FIVE

conclusıon

Through our work on this project, we were able to gain a deeper understanding of Java's class structures and various storage mechanisms, such as stack, queue, and others. As the project required us to work through a greater number of errors and details, we found ourselves having to rely more heavily on online resources, which ultimately allowed us to develop a stronger sense of how to access and utilize the necessary information effectively.

In addition to the specific technical skills we gained, the project also allowed us to develop a range of more general problem-solving skills. As we worked through the different issues we encountered, we had to think critically and creatively about potential solutions, experimenting with different approaches until we were able to find the most effective solution.

Overall, this project was a valuable learning experience for all involved, helping us to enhance our knowledge of Java programming and our ability to tackle complex coding challenges in the future.

REFERENCES

*https://stackoverflow.com/*

[*https://www.w3schools.com/*](https://www.w3schools.com/)

*https://www.geeksforgeeks.org/*

Code of the Project



