DOKUZ EYLUL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME1251 PROJECT BASED LEARNING – II

FINAL REPORT

PROJECT – II

NUMBERS

by

Ömer Batuhan Tunçer

Alim Ulaş

Rıdvan Özdemir

Lecturers

Tech.Asst. Şerife Yılmaz

Ezgi Demir

IZMIR

19.04.2019

CHAPTER ONE

PROGRESS DESCRIPTION

The aim of the project is to develop a game in which players try to reach the target number in limited time. Game is played with two players: human player and computer. A random target number between 100-999 is determined initially. Players will try to reach/approach this target number in 30 seconds. Players use five random small numbers and one big number. During this time, human makes calculations in his mind or with paper. Computer tries to reach target number by using random operations with randomly chosen numbers. When the time is up, players announce their result numbers. The player with the closest number explains their solution strategy and if it is true player gets the point.

CHAPTER TWO

TASK SUMMARY

2.2. Completed Tasks

Batuhan Tuncer: At the beginning of the project, all classes and screen was designed. After that, random numbers and timing was initialized. Then, answer was taken from user as a infix form and it was converted postfix form. At last, postfix form was calculated and computer AI was created.

Alim Ulaş: At the beginning of the project, all classes and screen was designed. After that, random numbers and timing was initialized. Then, answer was taken from user as a infix form and it was converted postfix form. At last, postfix form was calculated and computer AI was created.

Rıdvan Özdemir: At the beginning of the project, all classes and screen was designed. After that, random numbers and timing was initialized. Then, answer was taken from user as a infix form and it was converted postfix form. At last, postfix form was calculated and computer AI was created.

2.2. Incomplete Tasks: Reasons & Explanations

Batuhan Tuncer: There is no incompleted task.

Alim Ulaş: There is no incompleted task.

Rıdvan Özdemir: There is no incompleted task.

2.3. Additional Improvements

Batuhan Tuncer: Menu that you can use mouse on it, was added. Also a tutorial page that you can learn how to play this game. was added. As the last addition, three different game mods as easy, normal and hard were added.

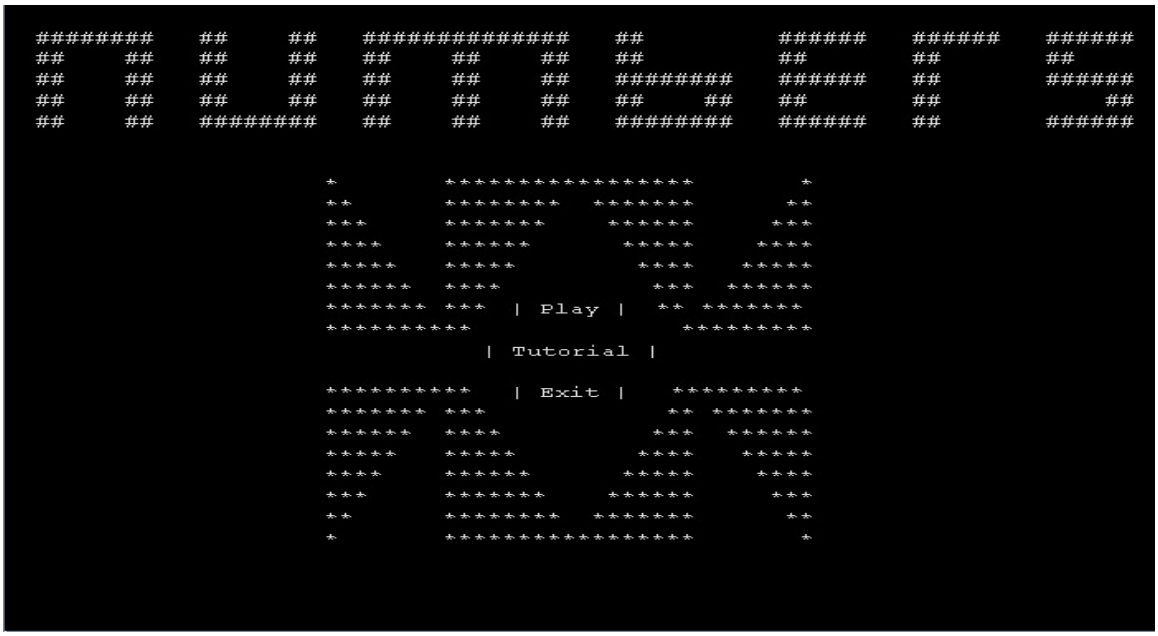
Alim Ulaş: Menu that you can use mouse on it, was added. Also a tutorial page that you can learn how to play this game. was added. As the last addition, three different game mods as easy, normal and hard were added.

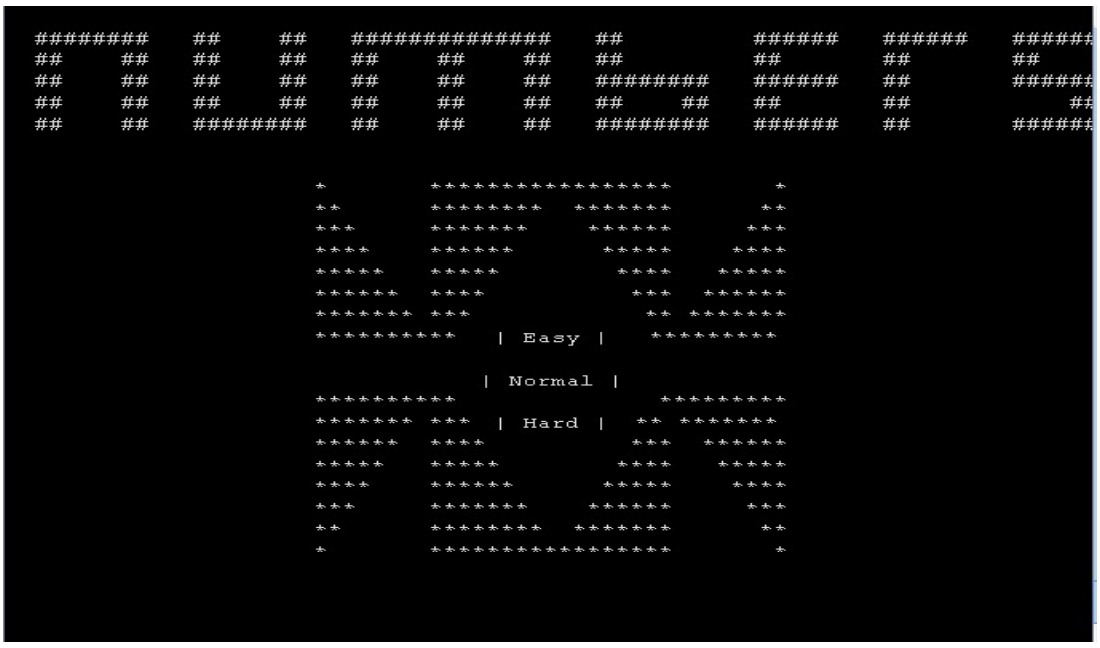
Rıdvan Özdemir: Menu that you can use mouse on it, was added. Also a tutorial page that you can learn how to play this game. was added. As the last addition, three different game mods as easy, normal and hard were added.

CHAPTER THREE

EXPLANATION OF ALGORITHM

3.1. Screenshots



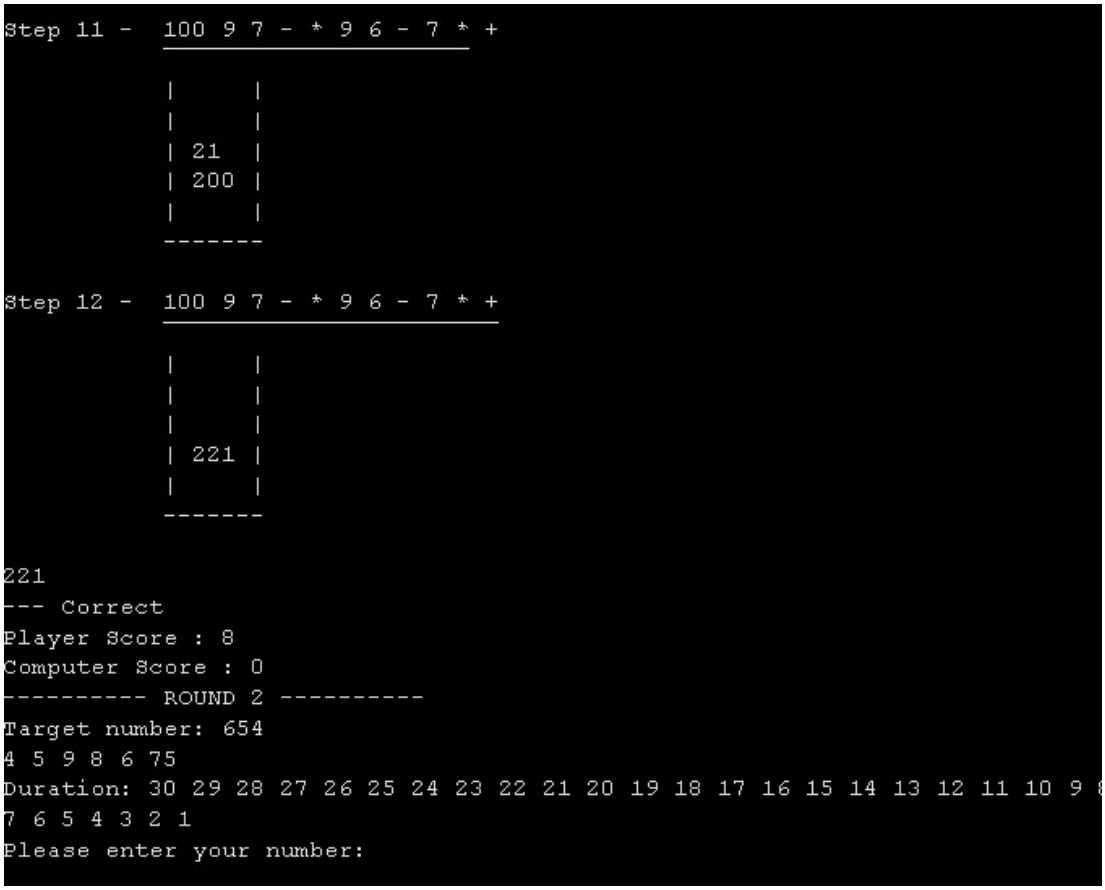

| Easy |
| Normal |
| Hard |

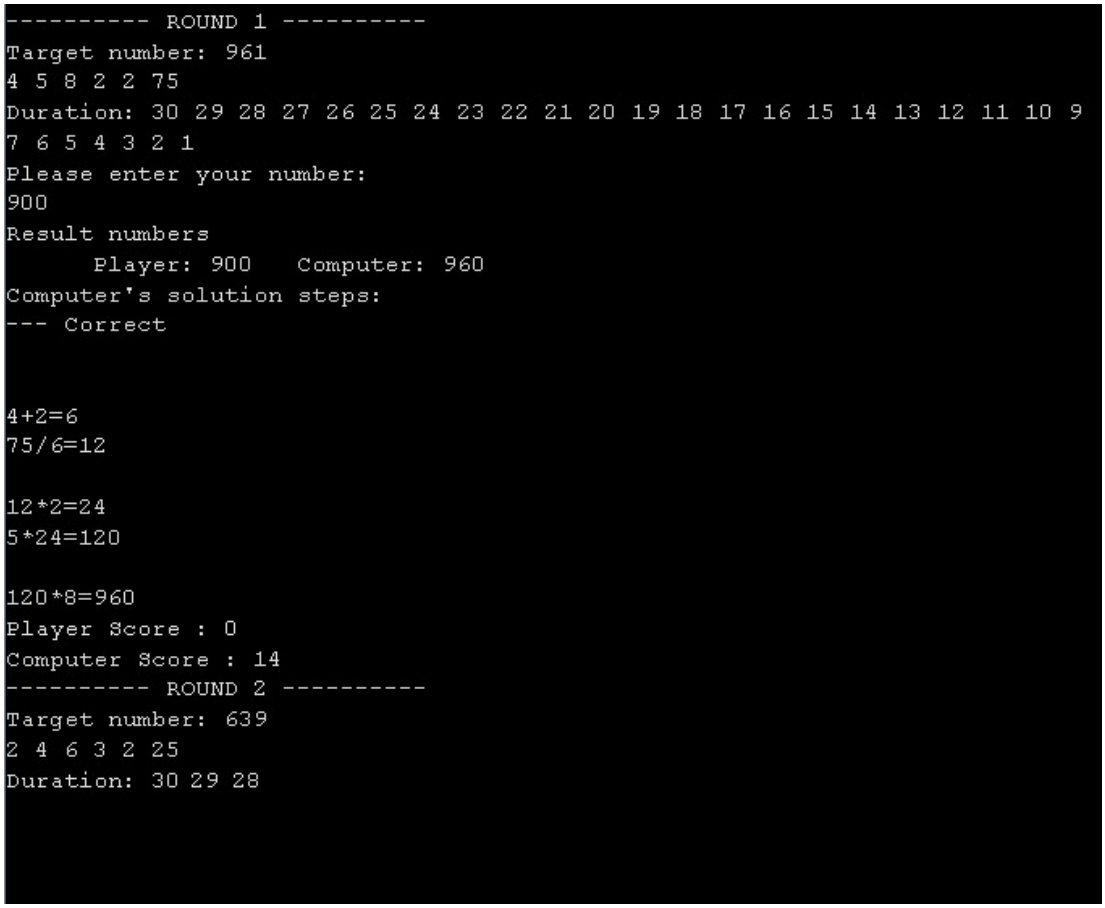

FIGHT AGAINST COMPUTER AI
A random target number between 100-999 is determined initially.
Players will try to reach this target number in 30 seconds. Players
use five random small numbers between 1-9, and one big number (25,50,
75 or 100) and 4 basic operations (* / + -) to reach this target.
GAME PLAYING RULES / STAGES
1. Target number is choosen randomly (100-999).
2. Five choosen number is choosen randomly (1-9).
3. One big number is choosen randomly (25/50/75/100).
4. 30-second countdown starts.
5. During this time, human player make calculations in his mind or.
with pen and paper.Computer tries to reach the target by using operations.
6. Rules for operations:
Players can use each number only once(given or calculated).
Players do not have to use all the numbers.
#. If parenthesis are not used, multiplication and division have higher
priority. Operations are calculated from left to right.
| MENU |
| NEXT |


---------- ROUND 1 ----------
Target number: 228
9 6 9 7 7 100
Duration: 30 29 28 27 26

```
Please enter your number:
221
Result numbers
      Player: 221   Computer: 221

please enter your Solve step
100*(9-7)+(9-6)*7
```

```
Step 6 - 100*(9-7)+(9-6)*7

 100 9

            |   |
            |   |
            | - |
            | ( |
            | * |
            -----
Step 7 - 100*(9-7)+(9-6)*7

 100 9 7

            |   |
            |   |
            | - |
            | ( |
            | * |
            -----
```


Step 11 - 100 9 7 - * 9 6 - 7 * +
| |
| |
| 21 |
| 200 |
| |
Step 12 - 100 9 7 - * 9 6 - 7 * +
| |
| |
| |
| 221 |
| |
221
--- Correct
Player Score : 8
Computer Score : 0
---------- ROUND 2 ----------
Target number: 654
4 5 9 8 6 75
Duration: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8
7 6 5 4 3 2 1
Please enter your number:



3.2. Functions

Conversion Function: This function converts infix form to postfix according to user's input.

Calculate Function: This function calculate postfix form.

Computer AI Function: This function tries to reach target number by using random operations with randomly chosen numbers.

3.3. Algorithm and Solution Strategies

Calculation and conversion classes were used specifically for the project. Stack of numbers and mathematical operators were created. These stacks enabled us to use postfix operations in order and to print on the screen. In this way, our operations were completed successfully. menu class was used for implementation of enigma in this project. System.currentTimeMillis() was used for the countdown. At this time implementation, start time and end time variables were used and the duration was printed according to difference of variables. For the calculation and conversion steps, if a number is found in the entered expression, this number is put into a string variable. If we have met the operator, this operator is put into a stack. If the incoming operator is '/' or '*' and if the last variable that put into stack is not the '(', the previous operator is put into a string variable. If the incoming operator is ')' , program adds statements to string expression until you see '(' operator in stack. If input is finished, all operators in the stack are added to the string expression. The string expression in the conversion process is controlled by "substring". If the expression is number, program checks the next expression is the character or space. then program take the last recorded number from the stack and did the operation. It was necessary to use the numbers we obtained as a result of the random use of numbers and operators to make the calculation of the computer. The process steps of the numbers we obtained should be kept in memory. That's why, a variable named solver was created. This variable kept the process steps of the numbers we obtained and result. Also, there is solver_checker command that ease our process. This variable was not able to used in stack or queue because stack and queue returns object. Two solution strategy was developed for this problem. First is using array and second is creating our own solverStack or solverQueue. Computer AI was created by using stack and array. Program always chose two numbers and did operations, also saved this operations steps. As a result of each transaction, the proximity of the result to the desired value was checked. The operation continues until the number in the array or the stack remains. Then, Then using the same numbers to do the same operation again. It continues for 30 seconds.

CHAPTER FOUR

PROBLEMS ENCOUNTERED

Batuhan Tuncer: The biggest problem I personally encountered was actually object-oriented programming. When I was working on the project, I sometimes forget what variable I had and where I kept it, and if I had to make a self-criticism for my own code, it might not be understood by someone else. We encountered the problem about user's input. Some type of the input was caused to crashing program, but they were fixed.

Alim Ulaş: I encountered the problem about the computer AI part of the project. At the core of this problem is putting the expression on the stack as a string. We encountered the memory error in the computer AI. We had a little problem about displaying conversion part.

Rıdvan Özdemir: Problem that I encountered is about the countdown part of the project. I used another system method for countdown part, but then I learned it has to be made by using System.currentTimeMillis(). We encountered the problem about user's input. Some type of the input was caused to crashing program, but they were fixed

CHAPTER FIVE

CONCLUSION

We completed all expected tasks of our project. We've learned the logic of stack and queue, how to convert infix form to postfix and calculate the postfix form and

REFERENCES

http://csis.pace.deu/~wolf/CS122/infix-postfix.htm

APPENDIX A

POSTER/WEB PAGE OF THE PROJECT