**DOKUZ EYLUL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CME1251 PROJECT BASED LEARNING – I**

**PROGRESS REPORT**

**PROJECT – I**

**SHAPER**

**by**

**Doğukan Dağlı**

**Ulaş Tepeli**

**Safa Nezir Özkan**

**Yunus Emre Sivri**

**Emirhan Çalışkan**

**Lecturers**

**Fatih Dicle**

**Şerife Yılmaz**

**21.10.2014**

# CHAPTER ONE

PROGRESS SUMMARY

## Work Done

Doğukan Dağlı : I completed the Week 3 task by implementing the C robot’s pathfinding logic using a DFS-like algorithm with a stack structure. I also integrated Ulaş’s snake eating algorithm with the full snake body structure. After that, I implemented the snake reverse feature, allowing the snake to reverse its tail correctly when stuck.

Ulaş Tepeli

I transferred the txt file to the screen, created the snake's eating algorithm and added a tail to the S robot and organized the movement of this tail.

Safa Nezir Özkan

The player's navigation on the map, how much movement per unit of time is added. Traps were added to the map and functions such as disappearing times and being released by pressing the space key were implemented. The general class skeleton was prepared. Timing loop was created. Emirhan's random input queue creation and Doğukan's functions of reading maze.txt and placing it on the board were implemented in the code. Added explosion of traps when snake is in range

*Yunus Emre Sivri*

*Emirhan Çalışkan*

*The S robots navigation(random and calculated) and movement(turnin,collision etc.) and input queue we use to put treasures to map.*

## Work in Progress

Doğukan Dağlı :I am currently working on refining the snake reversing mechanism, especially handling edge cases where the tail reverses direction too early or incorrectly.

Ulaş Tepeli.

I am working on the development of the eating algorithm

Safa Nezir Özkan

nothing actively involved.

Yunus Emre Sivri . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . .*(delete full stops and write your sentences)*

*Emirhan Çalışkan*

S robots collision with other S robots.

## Work to Be Done

Doğukan Dağlı I am planning to implement snake-to-snake collision mechanics, including scenarios like head-to-head collision, joining on hitting '1', and partial division on hitting '2' or '3'.

Ulaş Tepeli

S robots collision

SAfa Nezir Özkan

collisions will be worked on.

Yunus Emre Sivri . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . .*(delete full stops and write your sentences)*

Emirhan Çalışkan

Finishing S robots collisions with other S robots and implementation of it.

# CHAPTER TWO

algorıthms and solutıon strategıes

Doğukan Dağlı :For the C robot, I developed a pathfinding algorithm based on Depth First Search (DFS). I used a stack data structure to store the current path, and I continually searched for shorter paths in a loop of 500 iterations to optimize it.

Ulaş Tepeli.

The eating algorithm is based on the fact that if there are treasures on the moving part of the snake on the board, it saves them and updates the board. The algorithm for the movement of the snake's tail has two parts, first when the eating algorithm works, in which case the treasures recorded in the eating algorithm are added to the list that records the snake's tail, and second when the eating algorithm does not work and the snake just moves. In this case we use the list that holds the coordinates of the elements in the snake's tail. The algorithm in this section ends with each element of the list holding the coordinates taking and updating the coordinates of the previous element and refreshing the board.

Safa Nezir Özkan

When designing a time-dependent motion cycle, a variable called unit time was created and by modulating the multiples of this variable with the total time, it was tried to regulate which work would be done in how many seconds.

To solve the problem of the snake's tail being eaten by the player, the tail positions were checked in the same way as the wall and walkers positions.

Yunus Emre Sivri . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . .*(delete full stops and write your sentences & explain important algorithms and solution strategies in the project with code extracts. Use flowcharts if necessary)*

Emirhan Çalışkan

S robots have their own classes with all of the related information(position,direction,move counter etc.) and they are stored in an array that is in the main. Their position gets updated according to the game cycles.

# CHAPTER THREE

PROBLEMS ENCOUNTERED

Doğukan Dağlı :While implementing the snake reversing feature, I initially forgot to reverse the tail’s collected treasures list. As a result, after reversing, the snake body looked correct, but the logic for adding eaten treasures was broken. I fixed it by applying the same reversal logic used on the snake body to the collectibles list.

Ulaş Tepeli

While I was writing the eating algorithm, I encountered a problem such as not adding the saved treasure to the queue properly. I solved this by editing the add function in the single linked list class according to the project.

Sefa Nezir Özkan

When the traps were passed over, they were overwritten in the board variable and then disappeared. To solve this, I had to keep their positions in a position array and load this array back to the board each time.

Yunus Emre Sivri . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . *(delete full stops and write your sentences)*

*Emirhan Çalışkan*

When going in and out of random movement modes S robots would do incorrect movements.To solve this problem two different variables have been used to hold different modes directions.