DOKUZ EYLÜL ÜNİVERSİTESİ
İZMİR-1982

MÜHENDİSLİK FAKÜLTESİ
DOKUZ EYLÜL ÜNİVERSİTESİ

# WALLS AND MINES

## CME 1205 PROJECT BASED LEARNING-III

2022510127 KEREM KALINTAS
2020510158 ALI OZGUR INEP
2022510098 NISA AYDIN

DEUCENG
Dokuz Eylül University
Dept of Computer Engineering

## About The Game

The game is played in a game field including walls

The player is controlled by cursor keys

The aim of the game is to escape from enemies

# How we play

Follow along and have fun!

- By using the cursor keys,we can move the player.
- We can collect the numbers.Numbers have some functions such as energy,point,mine.
- To destroy the enemies,we can use mine.
- If the player has no energy,the movement speed is reduced by half.
- New enemies and numbers are added periodically.
- Walls are added or deleted periodically.

## REQUIREMENTS

- C# knowledge
- Algorithm knowledge
- Computer
- Teamwork

# TASK SHARING

Kerem KALINTAS

- GameLoop, Initialize functions
- Placing the enemies
- Adding and removing the walls
- Adding animation
- Editting the video

Nisa AYDIN

- Creating the walls
- Placing the numbers
- Preporing the presentation

Ali Ozgur INEP

- Progress and final report
- moveplayer,handleinput,set cell functions
- Chancing the color
- Setting the difficulty mode

## SCHEDULING

1.Week: Discussing the solution strategies, creating the walls, placing the human player.

2.Week: Human player movements,timing.

3. Week: Game initalization,placing the enemies and numbers.

4.Week: Adding or removing the walls,coloring the numbers,enemies and human player,starting a new game when pressing enter

5.Week: Adding explosion animation,preparing the presentation,writing the final report,editting the video

## COMPLETED TASKS

We have completed all the tasks.

## INCOMPLETE TASKS

There is no incomplate task.

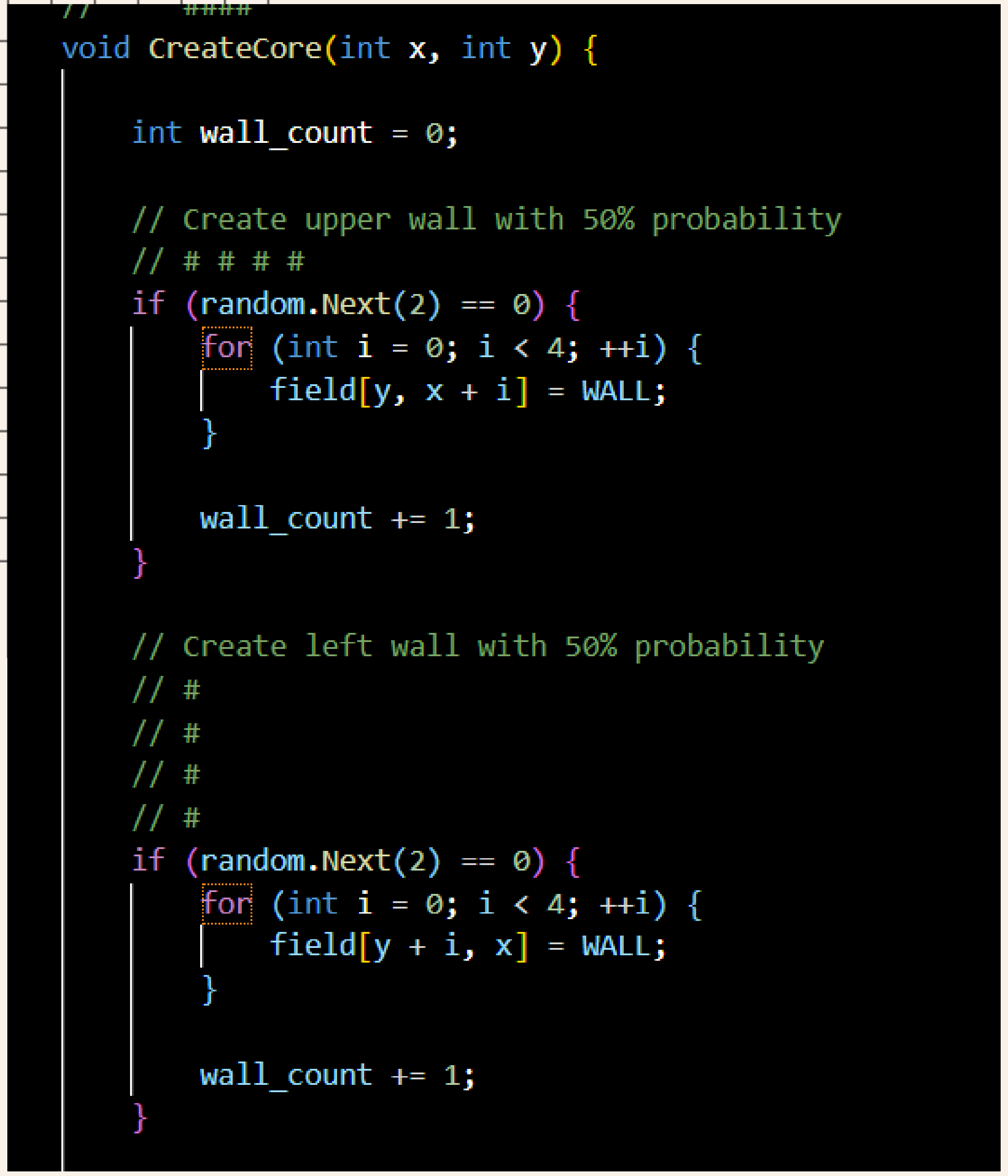
## ADDITIONAL IMPROVEMENTS

- The difficulty mode(easy,normal,hard)
- Changing the colours of the enemies ,numbers and the human player
- Adding the explosion animation

## PROBLEMS ENCOUNTERED

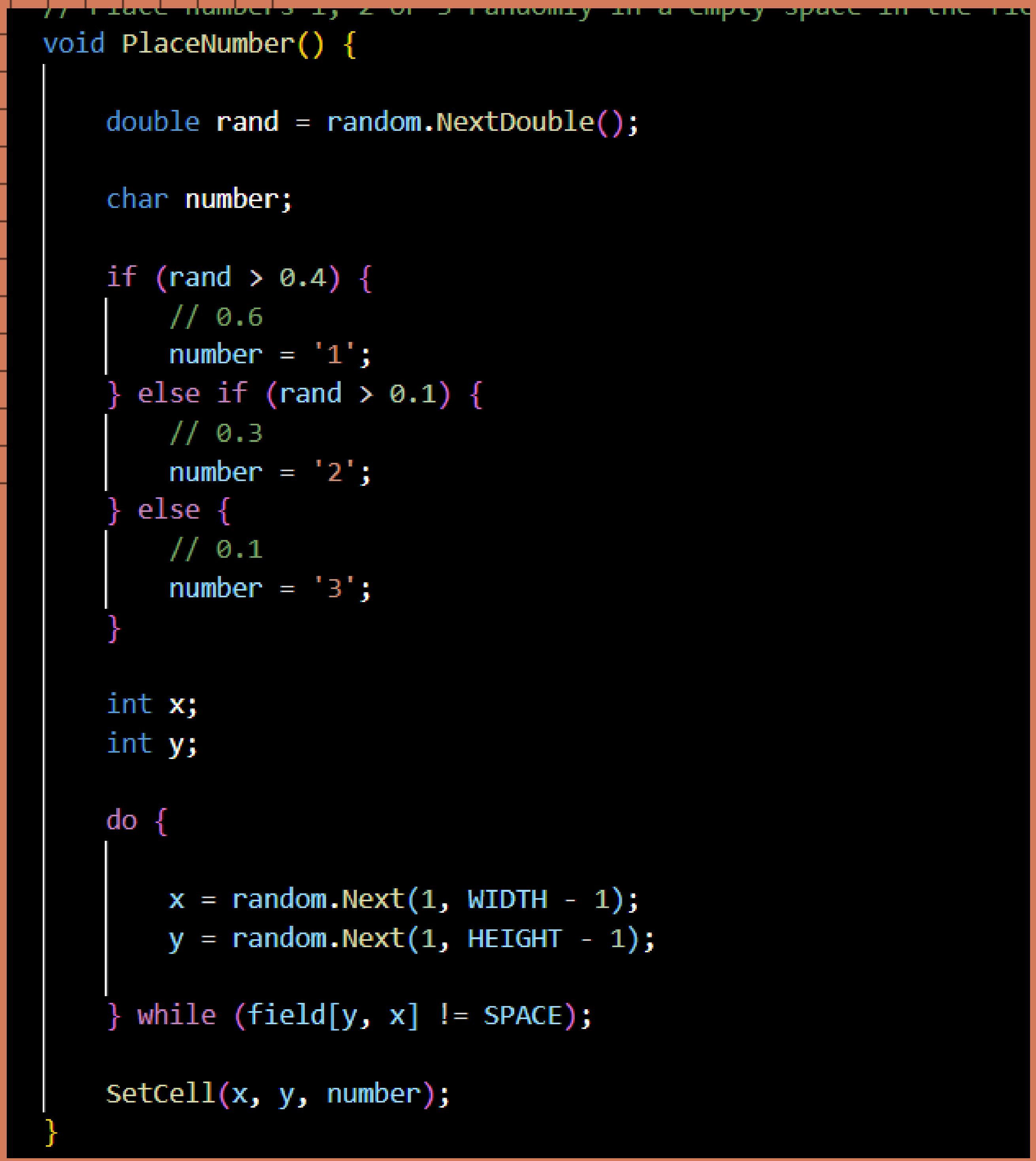
Mostly display errors occured because of the "SetCursorPosition".By making small calculations we handled this problem.

## ALGORITHMS AND SOLUTION STRATEGIES



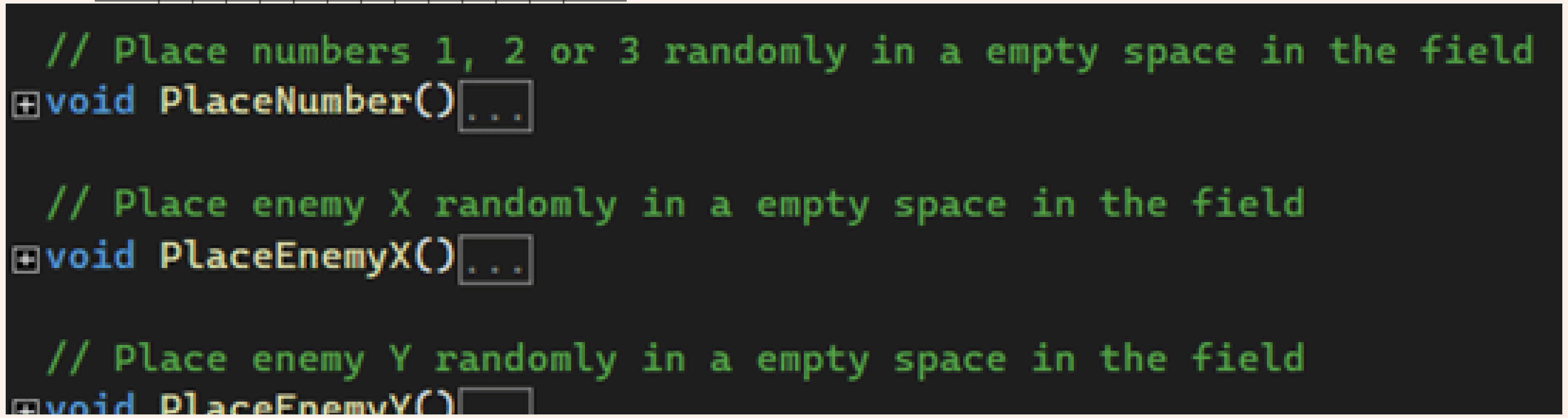
```
//      ####
void CreateCore(int x, int y) {

    int wall_count = 0;

    // Create upper wall with 50% probability
    // # # # #
    if (random.Next(2) == 0) {
        for (int i = 0; i < 4; ++i) {
            field[y, x + i] = WALL;
        }

        wall_count += 1;
    }

    // Create left wall with 50% probability
    // #
    // #
    // #
    // #
    if (random.Next(2) == 0) {
        for (int i = 0; i < 4; ++i) {
            field[y + i, x] = WALL;
        }

        wall_count += 1;
    }
```

# ALGORITHMS AND SOLUTION STRATEGIES



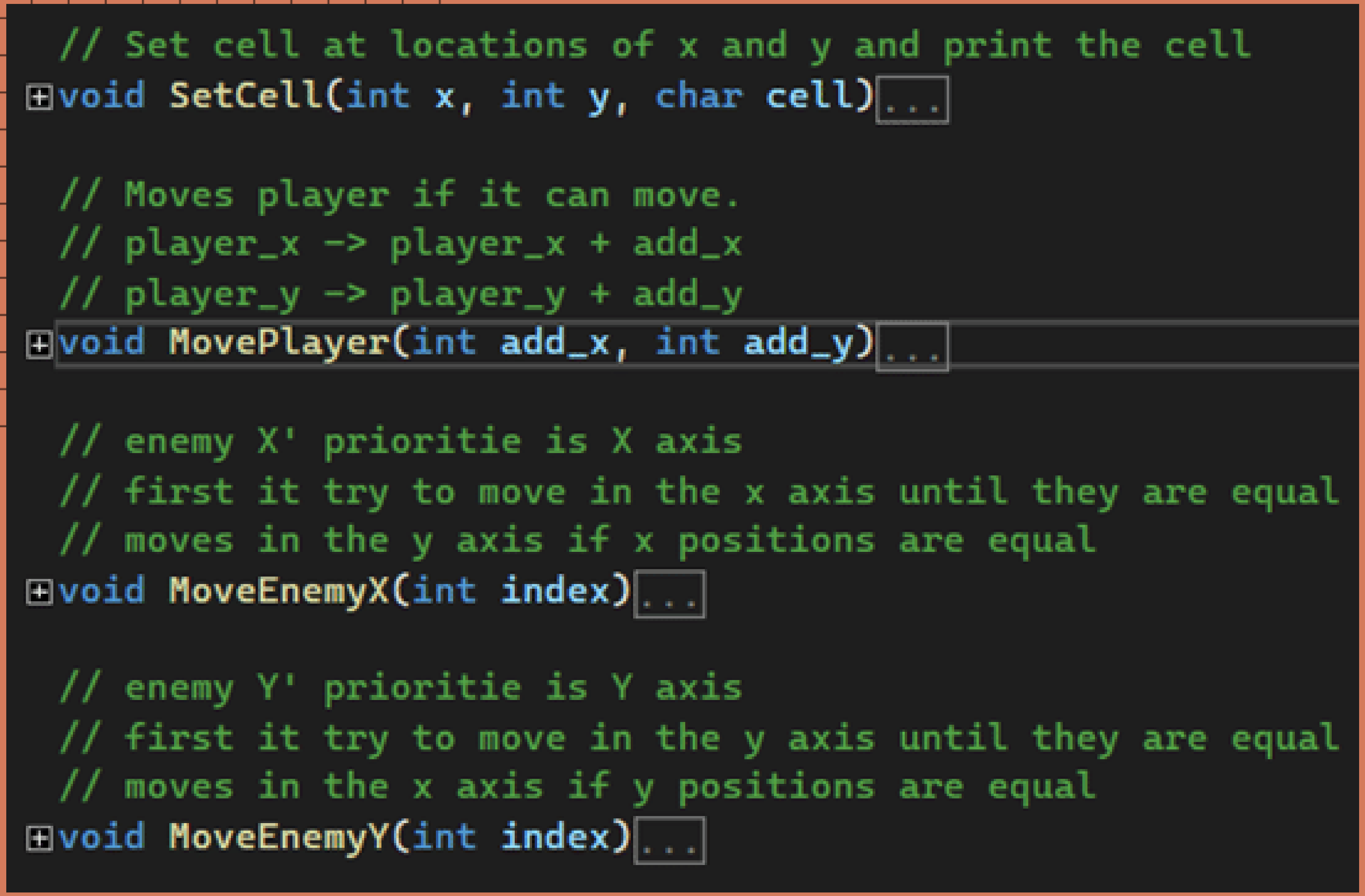
```
// Place numbers 1, 2 or 3 randomly in a empty space in the fie
void PlaceNumber() {

    double rand = random.NextDouble();

    char number;

    if (rand > 0.4) {
        // 0.6
        number = '1';
    } else if (rand > 0.1) {
        // 0.3
        number = '2';
    } else {
        // 0.1
        number = '3';
    }

    int x;
    int y;

    do {

        x = random.Next(1, WIDTH - 1);
        y = random.Next(1, HEIGHT - 1);

    } while (field[y, x] != SPACE);

    SetCell(x, y, number);
}
```

## ALGORITHMS AND SOLUTION STRATEGIES



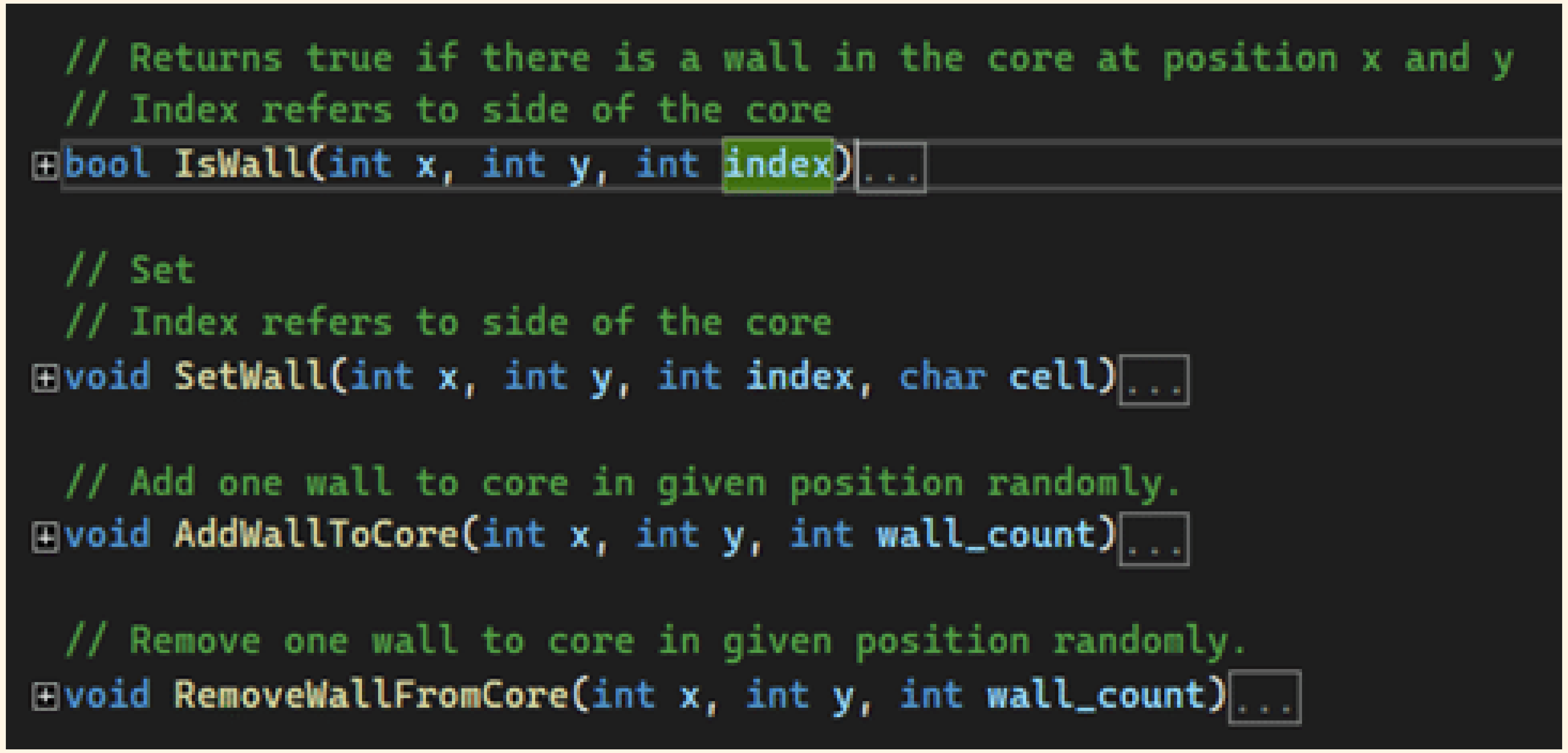
```
// Place numbers 1, 2 or 3 randomly in a empty space in the field
void PlaceNumber()...

// Place enemy X randomly in a empty space in the field
void PlaceEnemyX()...

// Place enemy Y randomly in a empty space in the field
void PlaceEnemyY()...
```

# ALGORITHMS AND SOLUTION STRATEGIES



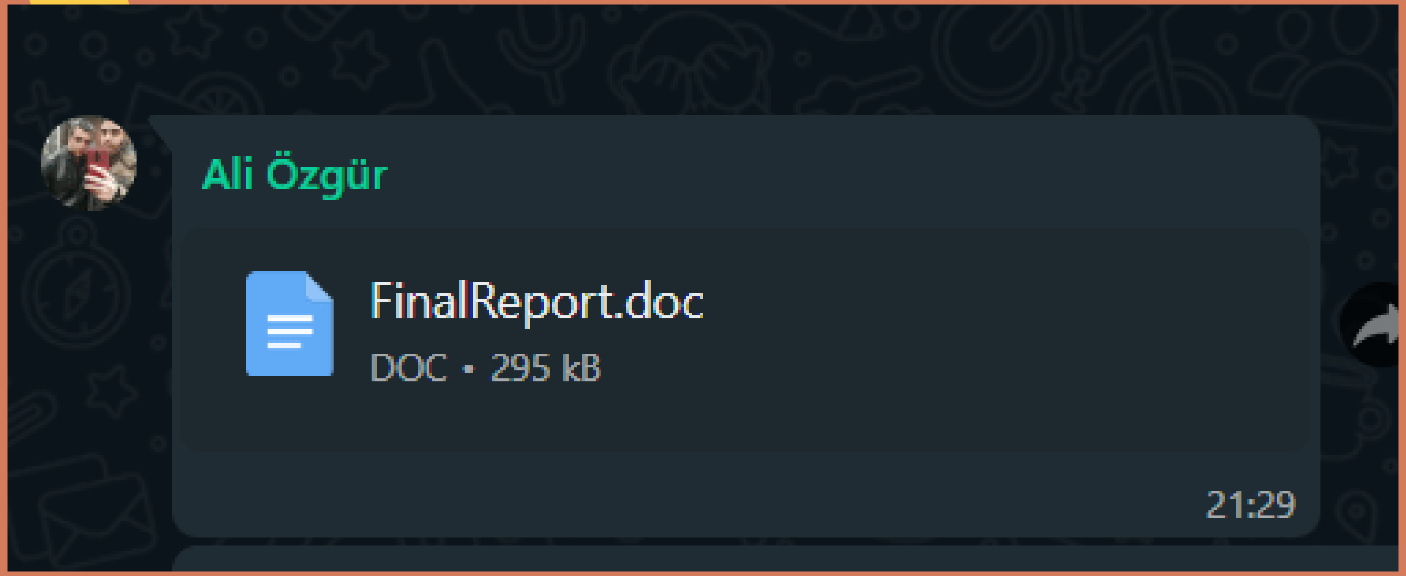
```
 // Set cell at locations of x and y and print the cell
⊞void SetCell(int x, int y, char cell)...

 // Moves player if it can move.
 // player_x -> player_x + add_x
 // player_y -> player_y + add_y
⊞void MovePlayer(int add_x, int add_y)...

 // enemy X' prioritie is X axis
 // first it try to move in the x axis until they are equal
 // moves in the y axis if x positions are equal
⊞void MoveEnemyX(int index)...

 // enemy Y' prioritie is Y axis
 // first it try to move in the y axis until they are equal
 // moves in the x axis if y positions are equal
⊞void MoveEnemyY(int index)...
```

## ALGORITHMS AND SOLUTION STRATEGIES



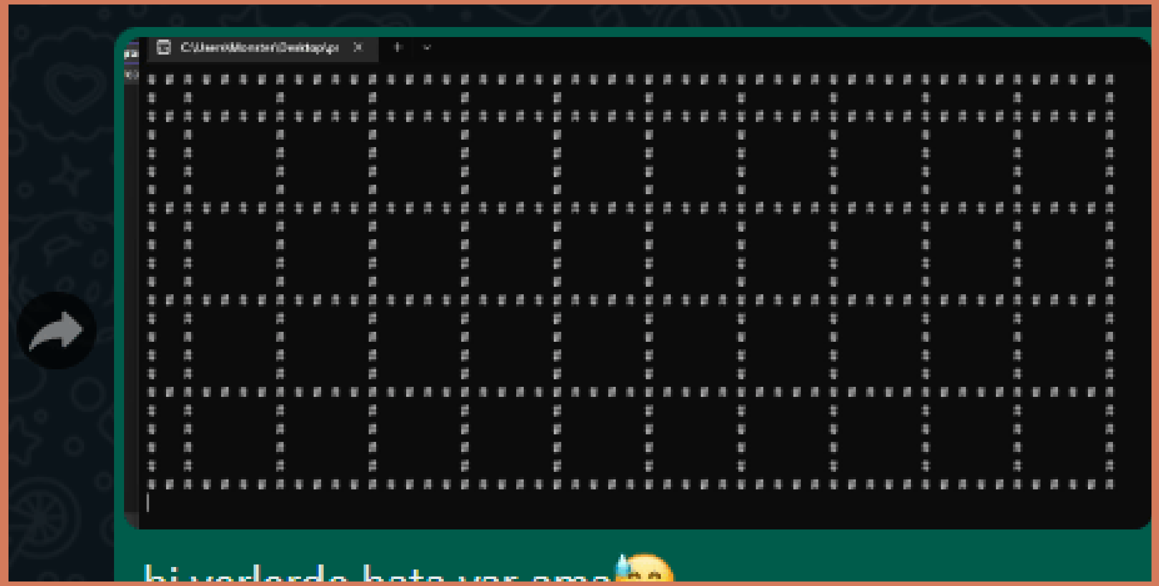
```
  // Returns true if there is a wall in the core at position x and y
  // Index refers to side of the core
⊞bool IsWall(int x, int y, int index)[...]

  // Set
  // Index refers to side of the core
⊞void SetWall(int x, int y, int index, char cell)[...]

  // Add one wall to core in given position randomly.
⊞void AddWallToCore(int x, int y, int wall_count)[...]

  // Remove one wall to core in given position randomly.
⊞void RemoveWallFromCore(int x, int y, int wall_count)[...]
```

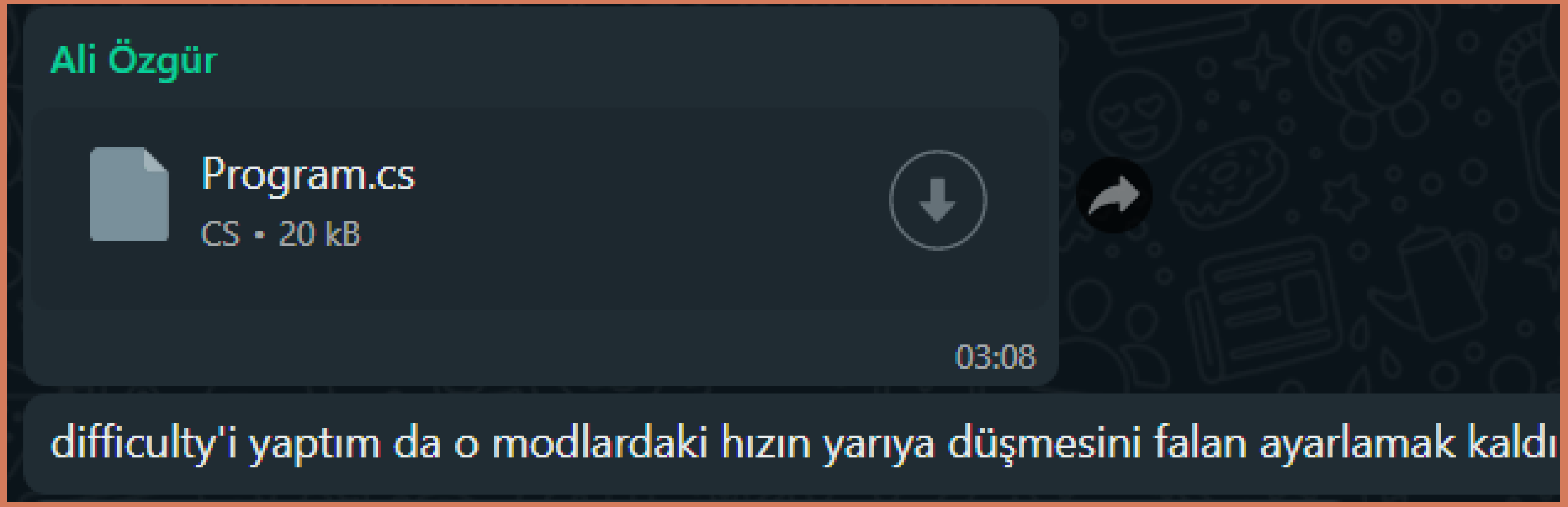
# SCREENSHOTS


Ali Özgür

FinalReport.doc
DOC • 295 kB

21:29


bi yerlerde hata var ama

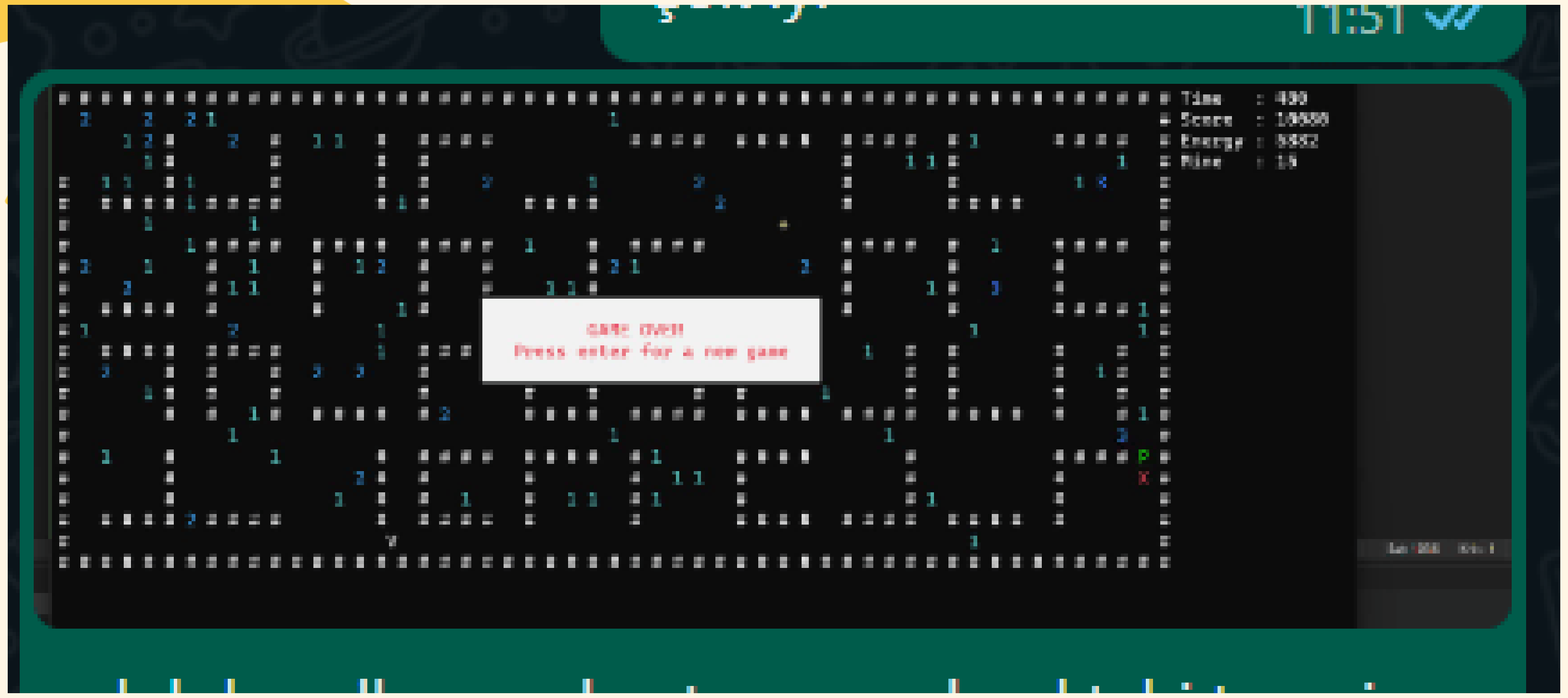

Ali Özgür

Program.cs
CS • 20 kB

03:08

difficulty'i yaptım da o modlardaki hızın yarıya düşmesini falan ayarlamak kaldı

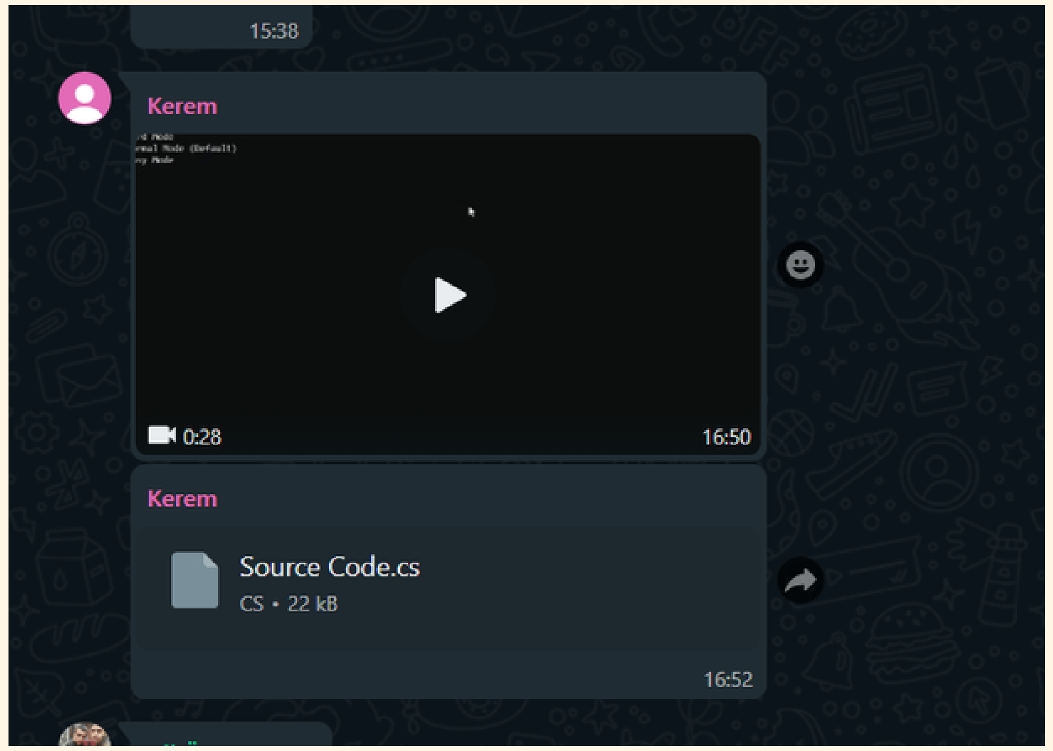
# SCREENSHOTS



11:51

```
Time   : 400
Score  : 10000
Energy : 8882
Mine   : 15
```

```
GAME OVER!!
Press enter for a new game
```



15:38

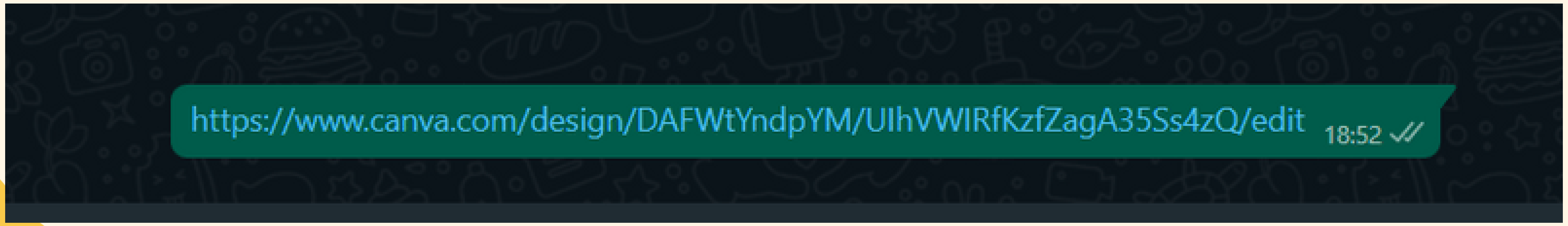
Kerem

0:28 16:50

Kerem

Source Code.cs
CS • 22 kB

16:52



https://www.canva.com/design/DAFWtYndpYM/UIhVWIRfKzfZagA35Ss4zQ/edit 18:52

## CONCULUSION

- In conculusion,we made a game which is similar to PacMan.
- The main thing is to work as a team on some transactions.
- We improved the sociality among us in a positive way while we made this project.

## REFERENCES

- For presentation

https://www.canva.com/design/DAFWtYndpYM/UlhVWlRfKzfZagA35Ss4zQ/edit

- To edit the video

https://online-video-cutter.com/

- Console.ReadKey Method

https://learn.microsoft.com/en-us/dotnet/api/system.console.readkey?view=net-7.0

- Math.Abs Method

https://learn.microsoft.com/en-us/dotnet/api/system.math.abs?view=net-7.0

- Console.KeyInfo Method

https://learn.microsoft.com/en-us/dotnet/api/system.consolekeyinfo?view=net-7.0

# Thank You