

## Proje raporu

1-ilk olarak kodumu başlangıç noktasını bulmak için for loopu kurdum bu onu 'X' olarak başlattım

```
30     for (int i = 0; i < map.length; i++) {
31         for (int j = 0; j < map[i].length; j++) {
32             if (map[i][j] == 'B') {
33                 konum[0] = i;
34                 konum[1] = j;
35                 map[i][j] = 'X';
36             }
37         }
38     }
```

2-olarak gamelsOver boolean tanımladım false olarak koudum ondan sonra while loop içine attım flase ise loop çalışmaya devam edecek

```
System.out.println(x: "Maze Game");
System.out.println("Moves: " + moves);
System.out.println(x: "Your Mark: ");
print1DArray(array: konum);
System.out.print("Click: W, A, S, D To move Or + \n"
    + "To Use A Bonus. To Exit press 'exit' .");
System.out.println(x: "");
System.out.println(x: "Avalibal bounese;");
System.out.println("F:" + bonusF + " R:" + bonusR + " T:" + bonusT + " H:" + bonusH);

print2DArray(array: map);

String girdi = scan.next();

if (girdi.equals(anObject: "+")) {
    useBonus(bonusH, bonusT, map, konum, moves);
    continue;
}
```

*3-olarak kullanıcıdan alınan W , A , S , D tuşlarını switch case ile yaptım*

*Ver her Tuş için methodlar hazırladım*

```
switch (girdi.toUpperCase()) {
    case "W":
        movePlayer(map, konum, rowChange:-1, colChange:0, bonusH, moves, bonusR, bonusF, bonusT, gameIsOver);
        bonusT += checkForBonusesT(map, konum, rowChange:-1, colChange:0, bonusH, moves, bonusR, bonusF, bonusT);
        bonusH += checkForBonusesH(map, konum, rowChange:-1, colChange:0, bonusH, moves, bonusR, bonusF, bonusT);
        bonusF += checkForBonusesF(map, konum, rowChange:-1, colChange:0, bonusH, moves, bonusR, bonusF, bonusT);
        bonusR += checkForBonusesR(map, konum, i:-1, rowChange:0, colChange:bonusH, moves, bonusR, bonusF, bonusT);
        gameIsOver = checkEnd(map, konum, rowChange:-1, colChange:0, gameIsOver);
        break;
    case "A":
        movePlayer(map, konum, rowChange:0, colChange:-1, bonusH, moves, bonusR, bonusF, bonusT, gameIsOver);
        bonusT += checkForBonusesT(map, konum, rowChange:0, colChange:-1, bonusH, moves, bonusR, bonusF, bonusT);
        bonusH += checkForBonusesH(map, konum, rowChange:0, colChange:-1, bonusH, moves, bonusR, bonusF, bonusT);
        bonusF += checkForBonusesF(map, konum, rowChange:0, colChange:-1, bonusH, moves, bonusR, bonusF, bonusT);
        bonusR += checkForBonusesR(map, konum, i:0, rowChange:-1, colChange:bonusH, moves, bonusR, bonusF, bonusT);
        gameIsOver = checkEnd(map, konum, rowChange:0, colChange:-1, gameIsOver);
        break;
    case "S":
        movePlayer(map, konum, rowChange:1, colChange:0, bonusH, moves, bonusR, bonusF, bonusT, gameIsOver);
        bonusT += checkForBonusesT(map, konum, rowChange:1, colChange:0, bonusH, moves, bonusR, bonusF, bonusT);
        bonusH += checkForBonusesH(map, konum, rowChange:1, colChange:0, bonusH, moves, bonusR, bonusF, bonusT);
        bonusF += checkForBonusesF(map, konum, rowChange:1, colChange:0, bonusH, moves, bonusR, bonusF, bonusT);
        bonusR += checkForBonusesR(map, konum, i:1, rowChange:0, colChange:bonusH, moves, bonusR, bonusF, bonusT);
        gameIsOver = checkEnd(map, konum, rowChange:1, colChange:0, gameIsOver);
        break;
}
```

*4-olarak harita ve konum arraylar oldukları için onları ekrana yazdırmak için 2 tane printarray methodlar yaptım*

```
public static void print2DArray(char[][] array) {
    for (int i = 0; i < array.length; i++) {
        for (int j = 0; j < array[i].length; j++) {
            System.out.print(array[i][j] + " ");
        }
        System.out.println();
    }
}

public static void print1DArray(int[] array) {
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i] + " ");
    }
    System.out.println();
}
```

*5-olarak mayınlar ve duvarlar için bir handelmine ve handelwall methodlar yaptım ve bu iki method direk oyuncunu*

*mayın ve duvar bonusları varsa*

```
public static void handleWall(char[][] map, int[] konum, int newRow, int newCol, int bonusR) {
    if (bonusR > 0) {
        map[newRow][newCol] = '.';
        konum[0] = newRow;
        konum[1] = newCol;
        bonusR--;
    } else {
        System.out.println("You cannot pass through the wall!");
    }
}

public static void handleMine(char[][] map, int[] konum, int newRow, int newCol, int bonusF, int moves) {
    if (bonusF > 0) {
        map[konum[0]][konum[1]] = '.';
        map[newRow][newCol] = 'X';
        konum[0] = newRow;
        konum[1] = newCol;
        bonusF--;
    } else {
        map[konum[0]][konum[1]] = '.';
        map[newRow][newCol] = 'X';
        konum[0] = newRow;
        konum[1] = newCol;
        moves += 5;
        System.out.println("Boom! Mine exploded. +5 moves.");
    }
}
```

*6-olarak her 5 hamleden sonra bonuslar  
random olarak haritada yerlerini  
değiştiriliyor*

*Onu math.random codu ile hazırladım*

```

public static char[][] RandomBomuses(char[][] map, int tempMoves) {
    for (int i = 0; i < map.length; i++) {
        for (int j = 0; j < map[i].length; j++) {
            if (tempMoves == 0 || tempMoves == 1 || tempMoves == 2 || tempMoves == 3 || tempMoves == 4 || tempMoves % 5 != 0) {
                break;
            }
            if (map[i][j] == 'T') {
                int ranRow = 0;
                int ranCol = 0;
                ranRow = (int) (Math.random() * map.length);
                ranCol = (int) (Math.random() * map[0].length);
                if (map[ranRow][ranCol] == '#' || map[ranRow][ranCol] == 'E' || map[ranRow][ranCol] == 'X' || map[ranRow][ranCol] == '.' || map[ranRow][ranCol] == 'H') {
                    map[i][j] = 'T';
                } else {
                    map[i][j] = '.';
                    map[ranRow][ranCol] = 'T';
                }
            }
            if (map[i][j] == 'H') {
                int ranRow = 0;
                int ranCol = 0;
                ranRow = (int) (Math.random() * map.length);
                ranCol = (int) (Math.random() * map[0].length);
                if (map[ranRow][ranCol] == '#' || map[ranRow][ranCol] == 'E' || map[ranRow][ranCol] == 'X' || map[ranRow][ranCol] == '.' || map[ranRow][ranCol] == 'H') {
                    map[i][j] = 'H';
                } else {
                    map[i][j] = '.';
                    map[ranRow][ranCol] = 'H';
                }
            }
        }
    }
}

```

*duvara, mayına yada 'E' yerine gelmemek şeklinde hazırladım*

*7- ve son olarak bir checkEnd methodu yaptım bu method haritada 'E' görünce gamelsOver booleane true olarak geri veriliyor*

```

private static boolean checkEnd(char[][] map, int[] konum, int rowChange, int colChange, boolean gameIsOver) {
    boolean End = gameIsOver;
    int newRow = konum[0] + rowChange;
    int newCol = konum[1] + colChange;
    if (newRow >= 0 && newRow < map.length && newCol >= 0 && newCol < map[0].length) {
        if (map[newRow][newCol] == 'E') {
            End = true;
        }
        else
            End = false;
    }

    return End;
}

```

*8-gamelsOver true olunca while loopten çıkıp*

```

        System.out.println(x: "=====");
    }
    System.out.println("You Won the Game With A total Of: " + moves + " moves!!!");
}

```

*bir message pirint ediliyor yanınada  
adım saysı veriliyor*

*Hamza Khouli*

*2321021368*

*Bilgisayar Mühendisliği / BLM22103 -  
Bilgisayar Programlama I (BLM22-G1)*