

Assignment 3

Report: Maze Game Code Review

- This report examines in detail how the given C program implements and runs a maze game. The program aims to navigate a player (symbolized by 'P') through a maze board to reach the end. The player is controlled using the W (up), A (left), S (down), and D (right) keys taken from the keyboard input.

Code Structure and Definitions :

- The code is written in C and utilizes standard input/output and time functions. The maze board is constructed with characters such as '#' (walls), '.' (empty spaces), 'P' (player), '1', and '2' (objects in the maze). The positions and values of objects in the maze are stored in predefined arrays. A series of functions manage the player's movements.

Game Initialization and Maze Generation

-The initialize_game() function determines randomly selected objects and tracks the used ones. The print_board() function displays the maze board on the screen and shows the player's current position and total earned ECTS.

Player Movements and Controls

-The move_player() function allows the player to move on the board according to keyboard inputs. The player can move without colliding with maze walls or objects, or exceeding the total ECTS acquired. After each move, the current state of the board and the player's new position are printed on the screen.

Main Game Loop and Termination Conditions

-The main function initializes the game within a loop and takes player moves. The player can end the game by pressing 'q' or complete the game by reaching the end of the maze. Upon game completion, the number of moves and total ECTS earned are displayed to the player.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #define size 16
5
6  int ones[8] = {119, 120, 121, 135, 137, 151, 152, 153};
7  int firstwall[16] = {102, 103, 104, 105, 106, 118, 122, 134, 138, 150, 154, 166, 167, 168, 169, 170};
8  int twowall[32] = {68, 69, 70, 71, 72, 73, 74, 75, 76, 84, 92, 100, 108, 116, 124, 132, 140, 148, 156, 164,
9  172, 180, 188, 196, 197, 198, 199, 200, 201, 202, 203, 204};
10 int twos[24] = {85, 86, 87, 88, 89, 90, 91, 107, 123, 139, 155, 171, 187, 186, 185, 184, 183, 182, 181,
11 165, 149, 133, 117, 101};
12 int selectedOnes[4];
13 int selectedTwos[3];
14 int usedOnes[8] = {0};
15 int usedTwos[24] = {0};
16 char board[size * size];
17 int ects = 0;
18 int movecount = 0;
19 int main_flag = 1;
20
```

```

64 void print_board()
65 {
66     initialize_game();
67     int k = 0;
68     int tmp_j;
69     for (int i = 1; i <= 16; i++)
70     {
71         if (i <= 4 || i >= 14)
72         {
73             for (int j = 1; j <= 16; j++)
74             {
75                 board[k] = '.';
76                 k++;
77             }
78         }
79         else if (i == 5 || i == 13)
80         {
81             for (int j = 1; j <= 16; j++)
82             {
83                 if (j >= 5 && j <= 13)
84                 {
85                     board[k] = '#';
86                     k++;
87                 }
88                 else
89                 {
90                     board[k] = '.';
91                     k++;
92                 }
93             }
94         }
95         else if (i == 6 || i == 12)
96         {
97             for (int j = 1; j <= 16; j++)
98             {
99                 if (j == 5 || j == 13)
100                 {
101                     board[k] = '#';
102                     k++;
103                 }
104                 else
105                 {
106                     board[k] = '.';
107                     k++;
108                 }
109             }
110         }
111     }

```

```

21 void initialize_game()
22 {
23     srand(time(NULL));
24
25     for (int i = 0; i < 4; i++)
26     {
27         int random_index;
28         do
29         {
30             random_index = rand() % 8;
31         } while (usedOnes[random_index]);
32         selectedOnes[i] = ones[random_index];
33         usedOnes[random_index] = 1;
34     }
35
36     for (int i = 0; i < 3; i++)
37     {
38         int random_index;
39         do
40         {
41             random_index = rand() % 24;
42         } while (usedTwos[random_index]);
43         selectedTwos[i] = twos[random_index];
44         usedTwos[random_index] = 1;
45     }
46
47     printf("Rastgele seçilen 4 ones eleman:\n");
48     for (int i = 0; i < 4; i++)
49     {
50         printf("%d\n", selectedOnes[i]);
51     }
52
53     printf("Rastgele seçilen 3 twos eleman:\n");
54     for (int i = 0; i < 3; i++)
55     {
56         printf("%d\n", selectedTwos[i]);
57     }
58 }
59
60
61
62
63

```

```

1 int main()
2 {
3     print_board();
4
5     while (main_flag)
6     {
7         char move;
8         printf("choice :");
9         scanf("%c", &move);
10        if (move == 'q')
11        {
12            main_flag = 0;
13            break;
14        }
15        else
16        {
17            move_player(move);
18        }
19    }
20 }

```

```

618     case 's':
619         movecount++;
620         last_index = current_index;
621         current_index += 16;
622         if (board[current_index] == '#')
623         {
624             if (board[current_index] == '#' && board[current_index + 16] == '#')
625             {
626             }
627             else
628             {
629                 for (int i = 0; i < 16; i++)
630                 {
631                     if (current_index == firstwall[i])
632                     {
633                         flag = 1;
634                         break;
635                     }
636                 }
637
638                 for (int i = 0; i < 32; i++)
639                 {
640                     if (current_index == twowall[i])
641                     {
642                         flag = 2;
643                         break;
644                     }
645                 }
646
647                 if (flag == 1 && ectcs >= 32)
648                 {
649                     current_index += 16;
650                     board[current_index] = 'P';
651                     board[last_index] = '.';
652                     printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
653                     for (int i = 0; i < 16 * 16; i++)
654                     {
655                         printf("%c ", board[i]);
656                         if ((i + 1) % 16 == 0)
657                         {
658                             printf("\n");
659                         }
660                     }
661                     printf("Total ECTS : %d\n", ectcs);
662                 }
663                 else if (flag == 2 && ectcs == 56)
664                 {
665                     current_index += 16;
666                     board[current_index] = 'P';
667                     board[last_index] = '.';
668                     printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
669                     for (int i = 0; i < 16 * 16; i++)
670                     {
671                         printf("%c ", board[i]);
672                         if ((i + 1) % 16 == 0)
673                         {
674                             printf("\n");
675                         }
676                     }
677                     printf("Total ECTS : %d\n", ectcs);
678                 }
679             }
680             else
681             {
682                 printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
683                 for (int i = 0; i < 16 * 16; i++)
684                 {
685                     printf("%c ", board[i]);
686                     if ((i + 1) % 16 == 0)
687                     {
688                         printf("\n");
689                     }
690                 }
691                 printf("Total ECTS : %d\n", ectcs);
692                 printf("Insufficient ectcs !!!\n");
693             }
694         }
695     }
696     else if (current_index == 255)
697     {
698         board[current_index] = 'P';
699         board[last_index] = '.';
700         printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
701         for (int i = 0; i < 16 * 16; i++)
702         {
703             printf("%c ", board[i]);
704             if ((i + 1) % 16 == 0)
705             {
706                 printf("\n");
707             }
708         }
709         printf("Total ECTS : %d\n", ectcs);
710         main_flag = 0;
711         printf("Congratulations! You have completed the game with %d moves and %d ECTS.\n\n", movecount, ectcs);
712     }
713     else if (board[current_index] == '1')
714     {
715         ectcs += 8;
716         board[current_index] = 'P';
717         board[last_index] = '.';
718         printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
719         for (int i = 0; i < 16 * 16; i++)
720         {
721             printf("%c ", board[i]);
722             if ((i + 1) % 16 == 0)
723             {
724                 printf("\n");
725             }
726         }
727         printf("Total ECTS : %d\n", ectcs);
728     }
729     else if (board[current_index] == '2')
730     {
731         ectcs += 8;
732         board[current_index] = 'P';

```

Youtube link :

<https://youtu.be/9iWal1CGAWo>