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.

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define size 16

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

```

```c
void print_board()
{
    initialize_game();
    int k = 0;
    int tmp_j;
    for (int i = 1; i <= 16; i++)
    {
        if (i <= 4 || i >= 14)
        {
            for (int j = 1; j <= 16; j++)
            {
                board[k] = '.';
                k++;
            }
        }
        else if (i == 5 || i == 13)
        {
            for (int j = 1; j <= 16; j++)
            {
                if (j >= 5 && j <= 13)
                {
                    board[k] = '#';
                    k++;
                }
                else
                {
                    board[k] = '.';
                    k++;
                }
            }
        }
        else if (i == 6 || i == 12)
        {
            for (int j = 1; j <= 16; j++)
            {
                if (j == 5 || j == 13)
                {
                    board[k] = '#';
                    k++;
                }
                else
                {
                    board[k] = '.';
                    k++;
                }
            }
        }
```

```c
void initialize_game()
{

    srand(time(NULL));

    for (int i = 0; i < 4; i++)
    {
        int random_index;
        do
        {
            random_index = rand() % 8;
        } while (usedOnes[random_index]);
        selectedOnes[i] = ones[random_index];
        usedOnes[random_index] = 1;
    }


    for (int i = 0; i < 3; i++)
    {
        int random_index;
        do
        {
            random_index = rand() % 24;
        } while (usedTwos[random_index]);
        selectedTwos[i] = twos[random_index];
        usedTwos[random_index] = 1;
    }


    printf("Rastgele seçilen 4 ones eleman:\n");
    for (int i = 0; i < 4; i++)
    {
        printf("%d\n", selectedOnes[i]);
    }


    printf("Rastgele seçilen 3 twos eleman:\n");
    for (int i = 0; i < 3; i++)
    {
        printf("%d\n", selectedTwos[i]);
    }
}
```

```c
int main()
{
    print_board();

    while (main_flag)
    {
        char move;
        printf("choice :");
        scanf("%c", &move);
        if (move == 'q')
        {
            main_flag = 0;
            break;
        }
        else
        {
            move_player(move);
        }
    }
}
```

```
    case 's':
        movecount++;
        last_index = current_index;
        current_index += 16;
        if (board[current_index] == '#')
        {
            if (board[current_index] == '#' && board[current_index + 16] == '#')
            {
            }
            else
            {
                for (int i = 0; i < 16; i++)
                {
                    if (current_index == firstwall[i])
                    {
                        flag = 1;
                        break;
                    }
                }

                for (int i = 0; i < 32; i++)
                {
                    if (current_index == twowall[i])
                    {
                        flag = 2;
                        break;
                    }
                }

                if (flag == 1 && ects >= 32)
                {
                    current_index += 16;
                    board[current_index] = 'P';
                    board[last_index] = '.';
                    printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
                    for (int i = 0; i < 16 * 16; i++)
                    {
                        printf("%c ", board[i]);
                        if ((i + 1) % 16 == 0)
                        {
                            printf("\n");
                        }
                    }
                    printf("Total ECTS : %d\n", ects);
                }
                else if (flag == 2 && ects == 56)
                {
                    current_index += 16;
                    board[current_index] = 'P';
                    board[last_index] = '.';
                    printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
                    for (int i = 0; i < 16 * 16; i++)
                    {
                        printf("%c ", board[i]);
                        if ((i + 1) % 16 == 0)
                        {
                            printf("\n");
                        }
                    }
                    printf("Total ECTS : %d\n", ects);
                }
                else
                {
                    printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
                    for (int i = 0; i < 16 * 16; i++)
                    {
                        printf("%c ", board[i]);
                        if ((i + 1) % 16 == 0)
                        {
                            printf("\n");
                        }
                    }
                    printf("Total ECTS : %d\n", ects);
                    printf("Insufficient ects !!!\n");
                }
            }
        }
        else if (current_index == 255)
        {
            board[current_index] = 'P';
            board[last_index] = '.';
            printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
            for (int i = 0; i < 16 * 16; i++)
            {
                printf("%c ", board[i]);
                if ((i + 1) % 16 == 0)
                {
                    printf("\n");
                }
            }
            printf("Total ECTS : %d\n", ects);
            main_flag = 0;
            printf("Congratulations! You have completed the game with %d moves and %d ECTS.\n\n", movecount, ects);
        }
        else if (board[current_index] == '1')
        {
            ects += 8;
            board[current_index] = 'P';
            board[last_index] = '.';
            printf("Use W (Up), A (Left), S (Down), D (Right) to move.\n");
            for (int i = 0; i < 16 * 16; i++)
            {
                printf("%c ", board[i]);
                if ((i + 1) % 16 == 0)
                {
                    printf("\n");
                }
            }
            printf("Total ECTS : %d\n", ects);
        }
        else if (board[current_index] == '2')
        {
            ects += 8;
            board[current_index] = 'P';
```

Youtube link :

https://youtu.be/9iWaI1CGAWo