**PROJECT REPORT**

Course Title (Course Code)



**BS(CS)-4A/B/C**

Group Members

|  |  |
| --- | --- |
| **Name** | **Enrollment** |
| 1. Nimra Mushtaq | 02-134202-018 |
| 1. Shahzadi Zainub | 02-134202-051 |
| 1. Tayyaba Imam | 02-134202-056 |

**Submitted to:**

**Saba Imtiaz**

**BAHRIA UNIVERSITY KARACHI CAMPUS**

Department of Computer Science

Module wise Work/Task Distribution

|  |  |  |
| --- | --- | --- |
| **Name** | **Enrollment** | **Task** |
| Nimra Mushtaq | 02-134202-018 | * Coding Maze functions * Project Report |
| Shahzadi Zainub | 02-134202-051 | * Coding Maze Design * Project Proposal |
| Tayyaba Imam | 02-134202-056 | * Coding Maze game structure and logic * Project Description and Working |

**TABLE OF CONTENT**

Contents

[CHAPTER 1 3](#_Toc94550017)

[INTRODUCTION 3](#_Toc94550018)

[1.1 Project Description 3](#_Toc94550019)

[1.2 Scope of the Project 3](#_Toc94550020)

[1.3 Modules in Project: 3](#_Toc94550021)

[1.4 Project Features 5](#_Toc94550022)

[CHAPTER 2 5](#_Toc94550023)

[REQUIREMENTS SPECIFICATION 5](#_Toc94550024)

[CHAPTER 3 6](#_Toc94550025)

[ANALYSIS 6](#_Toc94550026)

[3.1 Existing System 6](#_Toc94550027)

[3.2 Proposed System 6](#_Toc94550028)

[CHAPTER 4 6](#_Toc94550029)

[SYSTEM IMPLEMENTATION 6](#_Toc94550030)

[4.1 Introduction 6](#_Toc94550031)

[4.2 Code 7](#_Toc94550032)

[CHAPTER 5 14](#_Toc94550033)

[SAMPLE SCREEN SHOTS OF GAME 14](#_Toc94550034)

# CHAPTER 1

# INTRODUCTION

## 1.1 Project Description

This project is based on the design and implementation of a game called Maze Runner. Microsoft Visual C++ is used to create the game. In this game, the player must navigate a maze with three levels. Each level has a different level of difficulty, and the player must find his way through the game without colliding with any obstacles. The goal of the game is to navigate through the maze and complete the tasks required to advance to the final level.

## 1.2 Scope of the Project

Developing a game that incorporates data structures and algorithms is both difficult and rewarding. There are numerous data structures that can be incorporated, as well as numerous ways to programme the game. We're going to use Array to create a maze game named Maze Runner.

We will use comparison to check if the path ahead is valid If we have a possible path ahead, we will move ahead otherwise we print a message on screen as ‘Invalid move’. We can also move back and at any point, but we have only limited trials and once the trials are finished the game ends.

It will allow the user to navigate through the game and a character will indicate where the player current position is in the maze.

Maze shall also indicate the start and destination symbol.

To complete this game, you must navigate through a maze.

## 1.3 Modules in Project:

There are following modules used in the code

1. void printmaze(char maize[][25], int row, int col, int color);

This will print the maze by taking parameters such as the array, no of rows and columns and the color .

1. void goLeft(char[][25], int&startx, int&starty, int endx, int endy);

This function is used to move Left by taking parameters such as the array, starting index of rows and columns and the last index of rows and columns.

1. void goRight(char[][25], int&startx, int&starty, int endx, int endy);

This function is used to move Right by taking parameters such as the array, starting index of rows and columns and the last index of rows and columns.

1. void goUp(char[][25], int&, int&, int, int);

This function is used to move Up by taking parameters such as the array, starting index of rows and columns and the last index of rows and columns.

1. void goDown(char[][25], int&, int&, int, int);

This function is used to move Down by taking parameters such as the array, starting index of rows and columns and the last index of rows and columns.

1. void runmaze(char maize[][25], int x, int y, int winx, int winy, int max1, int max2, int color);

This function is used to print and enable movement around the maze it takes array the array index, the winning positions for x and y and the color to print the maze on screen.

## 1.4 Project Features

There are following features of the projects:

1. It implements a maze array made of characters
2. It asks for the user input using keys ‘A’,’D’,’S’,’W’ for Left, right, up, and down movement in the maze.
3. The user can choose any maze and the maze is printed on the screen.
4. The user can move around the maze to reach the position marked as ‘S’.
5. Once the user reaches the destination the game ends

The game ends.

1. It also has limited time and counts the invalid moves which should not be more than 3.

# CHAPTER 2

## REQUIREMENTS SPECIFICATION

We have used visual studio for C++ coding in our project.

Following requirements must be fulfilled:

1. Player can choose his favorite maze right from the start of the game.
2. Player can move in all four directions. Maze must keep updating with the current player location.
3. Player must finish the path quickly before the maximum number of moves available.
4. If the player makes an invalid move more than 3 times the game is over

# CHAPTER 3

## ANALYSIS

### 3.1 Existing System

The existing system makes use of arrays to implement a maze game. The array is of characters with symbols representing the player position the source and destination.

For a valid move the value of row or column in the array increases or decreases and replace the character with the asterisk ‘\*’ which will show that a move has been made.

### 3.2 Proposed System

We proposed the system to be implemented with stack and array where the stack will be useful if we want to backtrack through the maze. But it is not a requirement for this game and array is enough as well as efficient for successful implementation of a maze game.

# CHAPTER 4

## SYSTEM IMPLEMENTATION

### 4.1 Introduction

The Maze Game has been implemented using arrays. The array stores the original maze in the form of character array. The symbols “#” represents the walls and empty spaces is the valid path. The symbol ‘O’ is the location of player at start of the game and the symbol’S’ is the destination where the player must reach in time.

### 4.2 Code

#include<iostream>

#include <cstdlib> //To print in different colors

#include<windows.h>

#include<conio.h> //to accept input from keyboard

using namespace std;

//definition of global variables used in the code

int sec;

int maxtrial = 0;

int startx1 = 5, starty1 = 24, maxX1 = 23, maxY1 = 25, winx1 = 18, winy1 = 6;

int startx2 = 0, starty2 = 7, maxX2 = 20, maxY2 = 25, winx2 = 14, winy2 = 23;

int startx3 = 1, starty3 = 0, maxX3 = 25, maxY3 = 25, winx3 = 23, winy3 = 23;

char maze1[23][25] =

{

{ '#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#' },

{ '#',' ',' ',' ',' ','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ',' ','#',' ','#','#','#','#','#','#','#','#','#','#',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ',' ','#',' ','#',' ',' ',' ',' ',' ',' ',' ',' ','#',' ','#',' ',' ','#' },

{ '#','#','#','#','#','#',' ','#','#',' ','#',' ','#','#','#','#','#','#','#','#',' ','#',' ',' ','O' },

{ '#',' ',' ',' ',' ','#',' ','#',' ',' ','#',' ','#',' ',' ',' ',' ',' ',' ',' ',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ','#',' ',' ','#',' ','#','#','#','#','#',' ',' ',' ',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ','#','#','#','#',' ','#','#',' ',' ','#',' ',' ',' ',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ',' ',' ',' ',' ',' ','#','#',' ',' ','#',' ',' ',' ',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#','#','#','#','#','#','#','#','#',' ','#','#',' ',' ',' ',' ','#',' ',' ','#' },

{ '#','#','#','#','#','#',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#','#','#','#','#','#','#','#',' ','#','#','#','#','#','#',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ','#','#',' ',' ',' ',' ',' ','#',' ',' ',' ',' ','#',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ','#','#',' ','#','#','#','#','#','#',' ','#',' ','#',' ','#',' ',' ','#' },

{ '#','#','#','#','#','#',' ','#','#',' ','#','#','#','#','#','#',' ','#',' ','#',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#',' ',' ',' ',' ','#','#','#','#','#','#',' ','#','#','#',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#','#',' ','#',' ','#','#','#','#','#',' ',' ',' ',' ',' ',' ','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','S',' ',' ','#',' ','#',' ',' ',' ',' ',' ','#','#','#','#','#','#',' ',' ','#' },

{ '#',' ',' ',' ',' ','#','#','#','#',' ','#','#','#','#','#',' ',' ',' ',' ',' ',' ',' ',' ',' ','#' },

{ '#',' ',' ',' ',' ','#','#','#','#',' ','#','#','#','#','#','#','#','#','#','#','#','#','#',' ','#' },

{ '#',' ',' ',' ',' ','#',' ',' ',' ',' ','#',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','#' },

{ '#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#' }

};

char maze2[20][25] =

{

{ '#','#','#','#','#','#',' ','O',' ','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#' },

{ '#',' ','#',' ',' ','#',' ',' ','#',' ',' ',' ','#',' ',' ',' ',' ','#',' ',' ',' ',' ',' ','#','#' },

{ '#',' ','#',' ',' ','#',' ','#','#',' ','#',' ','#',' ','#','#',' ','#',' ','#','#','#',' ','#','#' },

{ '#',' ','#','#','#','#',' ',' ',' ',' ','#',' ',' ',' ',' ','#',' ','#',' ','#',' ',' ',' ','#','#' },

{ '#',' ','#',' ',' ','#','#','#','#','#','#','#','#','#',' ','#',' ','#',' ','#',' ','#','#','#','#' },

{ '#',' ','#',' ',' ','#',' ',' ',' ',' ',' ',' ',' ','#',' ','#',' ',' ',' ','#',' ',' ',' ','#','#' },

{ '#','#','#',' ',' ','#',' ','#','#','#','#','#',' ','#','#','#','#','#','#','#',' ','#',' ','#','#' },

{ '#',' ',' ',' ',' ','#',' ','#',' ',' ',' ','#',' ',' ',' ',' ',' ',' ',' ','#',' ','#',' ','#',' ' },

{ '#','#','#','#','#','#',' ','#','#','#',' ',' ',' ','#','#','#','#','#',' ','#','#','#',' ','#',' ' },

{ '#',' ',' ',' ',' ','#',' ',' ',' ','#','#','#','#','#',' ','#',' ',' ',' ',' ',' ',' ',' ','#',' ' },

{ '#',' ',' ',' ',' ','#',' ','#',' ',' ',' ',' ','#',' ',' ',' ',' ','#','#','#','#','#',' ','#',' ' },

{ '#',' ',' ',' ',' ','#',' ','#','#','#','#',' ','#','#','#','#','#','#',' ',' ',' ','#',' ','#',' ' },

{ '#',' ','#','#','#','#',' ',' ',' ','#',' ',' ','#',' ',' ',' ','#',' ',' ','#',' ','#',' ','#',' ' },

{ '#',' ','#',' ',' ','#',' ',' ',' ','#',' ','#','#',' ','#',' ','#',' ','#','#',' ','#','#','#','#' },

{ '#',' ','#',' ',' ',' ',' ',' ',' ','#',' ',' ',' ',' ','#',' ','#',' ',' ','#',' ',' ',' ',' ','S' },

{ '#',' ','#',' ',' ',' ',' ',' ','#','#','#','#','#','#','#',' ','#','#',' ','#','#','#','#','#','#' },

{ '#',' ','#',' ',' ',' ',' ','#',' ',' ',' ',' ',' ',' ',' ',' ',' ','#',' ','#',' ',' ',' ','#',' ' },

{ '#',' ','#',' ',' ','#',' ','#',' ','#','#','#','#','#','#','#','#','#',' ','#','#','#','#','#','#' },

{ '#',' ','#',' ',' ','#',' ','#',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','#','#' },

{ '#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#' }

};

char maze3[25][25] =

{

{ '#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#' },

{ '\*',' ','#',' ',' ',' ',' ',' ','#',' ',' ',' ',' ',' ',' ',' ','#',' ',' ',' ','#',' ',' ',' ','#' },

{ '#',' ','#','#','#','#','#',' ','#','#','#',' ','#',' ','#','#','#','#','#',' ','#',' ','#','#','#' },

{ '#',' ',' ',' ','#',' ',' ',' ',' ',' ',' ',' ','#',' ',' ',' ',' ',' ','#',' ','#',' ','#',' ','#' },

{ '#','#','#',' ','#',' ','#','#','#','#','#','#','#',' ','#','#','#',' ','#',' ','#',' ','#',' ','#' },

{ '#',' ',' ',' ','#',' ','#',' ',' ',' ',' ',' ','#',' ',' ',' ','#',' ',' ',' ',' ',' ','#',' ','#' },

{ '#',' ','#','#','#',' ','#',' ','#','#','#',' ','#','#','#','#','#','#','#','#','#',' ','#',' ','#' },

{ '#',' ','#',' ','#',' ','#',' ','#',' ','#',' ',' ',' ',' ',' ','#',' ',' ',' ','#',' ',' ',' ','#' },

{ '#',' ','#','#','#',' ','#',' ','#',' ','#',' ','#','#','#',' ','#',' ','#',' ','#','#','#',' ','#' },

{ '#',' ',' ',' ','#',' ','#',' ','#',' ','#',' ',' ',' ','#',' ','#',' ','#',' ',' ',' ',' ',' ','#' },

{ '#','#','#',' ','#',' ','#',' ','#',' ','#','#','#','#','#',' ','#','#','#',' ','#','#','#','#','#' },

{ '#',' ','#',' ','#',' ','#',' ',' ',' ','#',' ',' ',' ',' ',' ','#',' ',' ',' ',' ',' ',' ',' ','#' },

{ '#',' ','#',' ','#',' ','#',' ','#',' ','#',' ','#','#','#',' ','#','#','#',' ','#','#','#','#','#' },

{ '#',' ','#',' ',' ',' ','#',' ','#',' ','#',' ','#',' ',' ',' ',' ',' ',' ',' ',' ',' ','#',' ','#' },

{ '#',' ','#','#','#','#','#',' ','#',' ','#',' ','#','#','#','#','#','#','#',' ','#','#','#',' ','#' },

{ '#',' ',' ',' ','#',' ',' ',' ','#',' ','#',' ','#',' ',' ',' ',' ',' ','#',' ',' ',' ',' ',' ','#' },

{ '#',' ','#','#','#',' ','#','#','#',' ','#',' ','#',' ','#','#','#',' ','#',' ','#','#','#',' ','#' },

{ '#',' ',' ',' ','#',' ','#',' ',' ',' ','#',' ','#',' ','#',' ','#',' ','#',' ','#',' ','#',' ','#' },

{ '#','#','#',' ','#','#','#',' ','#','#','#',' ','#',' ','#',' ','#',' ','#','#','#',' ','#',' ','#' },

{ '#',' ','#',' ',' ','#','#',' ',' ',' ','#',' ','#',' ','#',' ','#',' ',' ',' ',' ',' ','#',' ','#' },

{ '#',' ','#','#','#',' ','#','#','#',' ','#',' ','#',' ','#',' ','#',' ','#','#','#',' ','#',' ','#' },

{ '#',' ','#',' ',' ',' ',' ',' ',' ',' ','#',' ',' ',' ','#',' ','#',' ','#',' ','#',' ','#',' ','#' },

{ '#',' ','#','#','#','#','#','#','#',' ','#','#','#',' ','#',' ','#','#','#',' ','#',' ','#','#','#' },

{ '#',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','#',' ',' ',' ',' ',' ',' ',' ','#',' ',' ',' ','S' },

{ '#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#' }

};

//Prototypes of functions used in the program

void printmaze(char[][25], int, int, int);

void goLeft(char[][25], int&, int&, int, int);

void goRight(char[][25], int&, int&, int, int);

void goUp(char[][25], int&, int&, int, int);

void goDown(char[][25], int&, int&, int, int);

void runmaze(char[][25], int, int, int, int, int, int, int);

//main function

int main()

{

int level;

//SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 5);

cout << "\nWelcome to the popular maze game";

cout << "\nGeneral instructions,use ";

cout << "\n w : to go up";

cout << "\n a : to go left";

cout << "\n d : to go right,and";

cout << "\n s: to go down. ";

cout << "\n\*\*\*\*\*\*\*\* Choose your LEVEL \*\*\*\*\*\*\*\*";

cout << "\nPlease choose the level you want to play";

cout << "\n Press 1 for Easy";

cout << "\n Press 2 for Medium";

cout << "\n Press 3 for Hard";

cout << "\n Press 4 to quit the game";

cout << "\nEnter your choice : ";

cin >> level;

// cases for the three different levels

switch (level)

{

case 1: {

sec = 70;

runmaze(maze1, startx1, starty1, winx1, winy1, maxX1, maxY1, 9); //function call

}

break;

case 2: {sec = 132;

runmaze(maze2, startx2, starty2, winx2, winy2, maxX2, maxY2, 9); //function call

}

break;

case 3: {sec = 150;

runmaze(maze3, startx3, starty3, winx3, winy3, maxX3, maxY3, 9); //function call

}

break;

case 4: cout << "\nGame ending...........\n";

}

//Back to original color

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 7);

system("pause");

return 0;

}

void printmaze(char maize[][25], int max1, int max2, int x) //function definition

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), x);

for (int i = 0; i < max1; i++)

{

for (int ii = 0; ii < max2; ii++)

{

cout << maize[i][ii];

}

cout << endl;

}

}

void goLeft(char maize[][25], int& x, int& y, int max1, int max2) //function definition

{

if ((maize[x][y - 1] == ' ' || maize[x][y - 1] == '-' || maize[x][y - 1] == '|' || maize[x][y - 1] == '\*') && (y - 1 > 0 || y - 1 == 0))

{

if (maize[x][y - 1] == ' ')

maize[x][y - 1] = '-';

else if (maize[x][y - 1] == '-')

maize[x][y - 1] = '\*';

else if (maize[x][y - 1] == '\*') {

maize[x][y - 1] = '-';

}

system("CLS");

printmaze(maize, max1, max2, 9);

cout << sec << endl;

sec--;

y = y - 1;

}

else

{

cout << "invalid move!";

maxtrial += 1;

}

}

void goRight(char maize[][25], int& x, int& y, int max1, int max2) //function definition

{

if ((maize[x][y + 1] == ' ' || maize[x][y + 1] == '-' || maize[x][y + 1] == '|' || maize[x][y + 1] == '\*') && (y + 1 < max2 - 1 || y + 1 == max2 - 1))

{

if (maize[x][y + 1] == ' ')

maize[x][y + 1] = '-';

else if (maize[x][y + 1] == '-')

maize[x][y + 1] = '\*';

else if (maize[x][y + 1] == '\*')

maize[x][y + 1] = '-';

system("CLS");

printmaze(maize, max1, max2, 9);

cout << sec << endl;

sec--;

y = y + 1;

if (sec < 0)

cout << "\nTime is up!Game over!";

}

else

{

cout << "invalid move!";

maxtrial += 1;

}

}

void goUp(char maize[][25], int& x, int& y, int max1, int max2) //function definition

{

if ((maize[x - 1][y] == ' ' || maize[x - 1][y] == '|' || maize[x - 1][y] == '-' || maize[x - 1][y] == '\*') && (x - 1 > 0 || x - 1 == 0))

{

if (maize[x - 1][y] == ' ')

maize[x - 1][y] = '|';

else if (maize[x - 1][y] == '|')

maize[x - 1][y] = '\*';

else if (maize[x - 1][y] == '\*')

maize[x - 1][y] = '|';

system("CLS");

printmaze(maize, max1, max2, 9);

cout << sec << endl;

sec--;

x = x - 1;

}

else

{

cout << "invalid move!";

maxtrial += 1;

}

}

void goDown(char maize[][25], int& x, int& y, int max1, int max2) //function definition

{

if ((maize[x + 1][y] == ' ' || maize[x + 1][y] == '|' || maize[x + 1][y] == '-' || maize[x + 1][y] == '\*') && (x + 1 == max1 - 1 || x + 1 < max1 - 1))

{

if (maize[x + 1][y] == ' ')

maize[x + 1][y] = '|';

else if (maize[x + 1][y] == '|')

maize[x + 1][y] = '\*';

else if (maize[x + 1][y] == '\*')

maize[x + 1][y] = '|';

system("CLS");

printmaze(maize, max1, max2, 9);

cout << sec << endl;

sec--;

x = x + 1;

}

else

{

cout << "invalid move!";

maxtrial += 1;

}

}

void runmaze(char maize[][25], int x, int y, int endx, int endy, int max1, int max2, int color) //function definition

{

printmaze(maize, max1, max2, color);

while ((x != endx || y != endy) && maxtrial < 3 && (sec > 0))

{

char move;

move = \_getch();

if (move == 'W' || move == 'w')

goUp(maize, x, y, max1, max2);

else if (move == 'S' || move == 's')

goDown(maize, x, y, max1, max2);

else if (move == 'A' || move == 'a')

goLeft(maize, x, y, max1, max2);

else if (move == 'D' || move == 'd')

goRight(maize, x, y, max1, max2);

else

cout << "invalid input" << endl;

}

if (maxtrial != 3 && sec > 0)

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 6);

cout << "\ncongradulations,you found your way through!" << endl;

}

else

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 4);

cout << "\nGame over!!! loser!" << endl;

}

}

# CHAPTER 5

## SAMPLE SCREEN SHOTS OF GAME

Graphical user interface

Description automatically generated

Graphical user interface, text

Description automatically generated

A picture containing graphical user interface

Description automatically generated



