

Data Structures & Algorithms Lab

(Fall-2024)

PROJECT PROPOSAL

Teacher: Engr. Raheela Ambrin

Date :26/Nov/2024



Bahria University, Islamabad
Department of Software Engineering

➤ **Project Title:**

Console Based Game: "Maze Runner"



➤ **Group Members:**

Members	Name
Mudassir Riaz	01-131232-046
M. Awais Ishaq	01-131232-052
M. Sami Shahid	01-131232-066

➤ Project Statement:

The project is a **Maze Runner Game** where players must navigate through a maze from a starting point (S) to an end point (E) while avoiding walls. The game will feature a **Time Attack mode**, where players have a limited amount of time to solve the maze and reach the exit. The game will also include an **AI Mode**, where an AI will attempt to solve the maze on its own using algorithms like BFS or DFS. Players can compare their performance to the AI's, adding an additional layer of challenge and learning. The goal is to reach the end as quickly as possible while making efficient moves. The game will be implemented using core data structures like stacks, queues, and graphs to handle player movement and solve the maze. The motivation behind this project is to practice implementing data structures and algorithms while providing an interactive, fun experience for players.

➤ Objectives:

1. To implement core C++ and DSA concepts in game development.
2. A well-designed, interactive and brainstorming game.
3. To develop a console-based game that is both engaging and instructive.
4. To enhance our teamwork, programming skills, and algorithmic thinking by different solving various problems that appear throughout the project.

➤ Technologies:

• **Programming Language:**

C++ will be used for the development of the maze runner game, chosen for its efficiency and ability to handle complex data structures.

• **Frameworks/Libraries:**

STL (Standard Template Library) for managing basic data structures like stacks, queues, and vectors.

• **Tools:**

Code Editor/IDE: Visual Studio Community.

Version Control: GitHub for version control and collaboration.

• **Data Structures and Algorithms:**

1. **Stack:** For implementing backtracking in the maze.

2. **Queue:** For implementing BFS (Breadth-First Search) Algorithm.
3. **Pairs:** Pairs representing the maze cells, where each pair (x, y) corresponds to a specific position in the maze.
4. **Algorithms:** DFS or BFS for solving the maze and guiding the player to the end.

➤ Features:

1. Time Attack Mode:

Players will have a limited time to navigate the maze and reach the exit (E). A timer will start as soon as the player begins their journey. The game will track the time it takes for the player to solve the maze, and players can compete to solve the maze as quickly as possible. The faster the player reaches the end, the higher their score.

2. AI Mode:

In **AI Mode**, an AI will autonomously navigate the maze from the start (S) to the end (E) using various path-finding algorithms. Players can select the algorithm to be used and watch the AI solve the maze. The algorithms implemented include:

- **Best First Search:** The AI uses heuristics to prioritize cells that appear to lead towards the exit, potentially solving the maze efficiently.
- **Breadth First Search (BFS):** The AI explores the maze layer by layer, ensuring that it finds the shortest path to the exit.
- **Depth First Search (DFS):** The AI explores the maze by going deeper into paths until it hits a dead-end, backtracking when necessary.
- **Dijkstra Search:** The AI calculates the shortest path to the exit by considering the least-cost steps to reach each cell.

Players can compare their performance with the AIs, providing an opportunity to analyze the efficiency of each algorithm. The AI's path-finding efficiency and the number of cells it visits will vary based on the chosen algorithm.

Conclusion:

This console-based game will be an exciting opportunity for us to apply programming and DSA skills and to develop a meaningful project. The game will showcase our understanding of C++ and demonstrate our ability to work effectively as a team while tackling real-world programming challenges that we face throughout our project.

