

TITLE: DUNGEON ESCAPE

# C-MAJOR PROJECT

# ABSTRACT

---

- ✓ THIS PROJECT IMPLEMENTS A USER BASED GAME IN WHICH THE USER CHOOSES THE RIGHT PATH TO ESCAPE THE DUNGEON . WHILE READING USER INPUT . IT DEMONSTRATE THE USE OF THE MODULAR PROGRAMMING TO BUILD A SCALABE AND MAINTAINABLE SOFTWARE SYSTEM.

# PROBLEM DEFINITION

---

1. A maze is Generated
2. The player starts at position(1,1)
3. The exit is located at bottom right of the maze
4. Player can be moved using w\l\a\s\d
5. The game ends when :
  - The player exits the maze.

# SYSTEM DESIGN AND ALGORITHM

- The system is modular consisting of source file:  
• “game.c” checks for win condition and player movement

- “main.c” shows main
- “maze.c” generates maze
- “player.c” handles movement checks for collision

Algorithm:

1. Generates maze
2. Reads movement
3. Checks for collisions
4. Checks exit

# IMPLEMENTATION

---

Key language features used:

Data Types- int, character, extern keyword

Functions – for modularity and clarity

# TESTING AND RESULTS

1. Case1: Manual movement through terminal leads to a complex movement after many steps , user either exits or wins.
2. Case2: Using a ./main < sample\_input.txt file that has a predefined sets of moves and leads to graceful execution.
3. Case3: Using manual input.txt file for right set of moves and lose conditons.

RESULTS:

✓ Win condition:

```
#####
#...#....#
#.#.#.###..#
#.#. .... #P.#
#. #####.##.#
#. .... #...#
#####
```

You Escaped the Dungeon!

✓ Lose condition:

```
#####
#...#. P.#
#.#.#.###..#
#.#. .... #X.#
#. #####.##.#
#. .... #...#
#####
```

Game Over!

# CONCLUSION

---

The project achieves its goal of creating a functional dungeon escape game using a modular C program. Player movement, boundary checks, and win detection operate correctly, confirming a stable and well-organized implementation.