Class: BSCS-3B
Year: 2025 Fall

BAHRIA UNIVERSITY KARACHI

DEPARTMENT OF COMPUTER SCIENCE

Data Structure & Algorithm Lab
(1 Credit Hour)
CSL-221

Report

Class: BSCS-3B
Year: 2025 Fall

Data Structure & Algorithm Lab
(1 Credit Hour)

Group Members:
Wardha Khalid(02-134242-096)
Maheen Fatima(02-134242-011)
Ayyan Khan(02-134242-015)

Class & Section: *BSCS-3B*

Semester: *3rd Semester*

Fall / spring: *Fall-2025*

Lab Day & Time: *Thursday 8:30-11:30 AM*

Course Teacher: *Rabia Amjad*

Assignmnet Submission: *12December'2025*

Class: BSCS-3B
Year: 2025 Fall

Project Report: Console-Based Snake Game Using Doubly Linked List in C++

Table of Contents

Class: BSCS-3B
Year: 2025 Fall

Acknowledgement

I would like to thank **Engr. Rabia Amjad** for her continuous guidance and valuable feedback throughout this project. Her insights into C++ programming and data structures helped me implement the Snake Game efficiently.

Abstract

This project presents the implementation of a **console-based Snake Game in C++**, utilizing a **doubly linked list** to represent the snake's body. The project demonstrates key programming concepts such as **object-oriented design**, **dynamic memory management**, and **file handling**. The game includes a functional **menu**, **leaderboard**, **real-time keyboard control**, **collision detection**, and **random food generation**. It serves as an application of fundamental data structure principles in a fun and interactive manner.

Introduction

The Snake Game is a classic arcade-style game where a player controls a snake that grows longer by eating food and loses if it collides with itself or the wall. The objective is to achieve the highest possible score. This project replicates this logic using **C++** and demonstrates how core programming concepts are applied in a real-world scenario.

Class: BSCS-3B
Year: 2025 Fall

Problem Statement

To design and implement a **console-based Snake Game** that: - Uses a **doubly linked list** to manage the snake's body dynamically. - Supports **real-time movement and control**. - Detects **collisions** with walls and the snake itself. - Generates **random food** that the snake can consume. - Maintains a **leaderboard** using file storage.

Objectives

- Apply **data structure** and **object-oriented programming** concepts.
- Use **file I/O** for persistent data (leaderboard).
- Implement real-time user interaction in a **console environment**.
- Design modular, readable, and maintainable code.

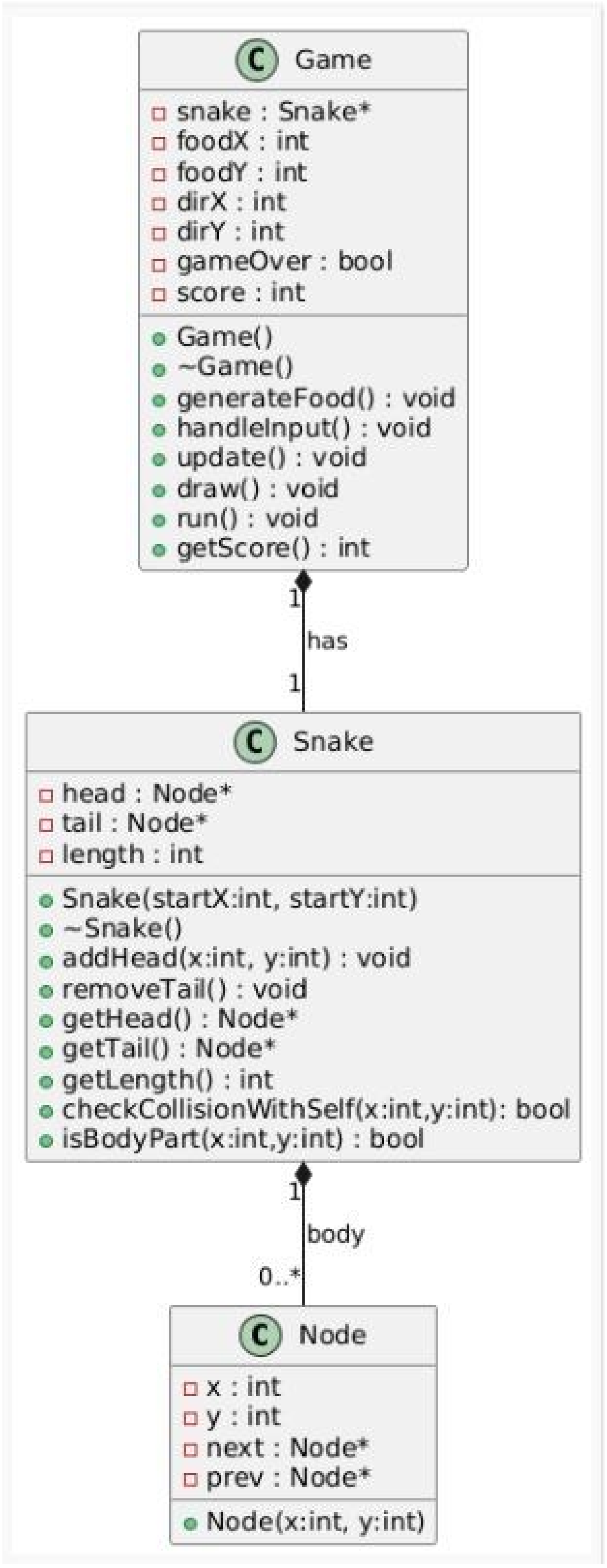
Scope

In Scope: - Single-player console-based gameplay. - Real-time movement using keyboard inputs. - Persistent leaderboard.
Out of Scope: - Graphical interface or animation. - Multiplayer functionality. - Complex levels or obstacles.

Class: BSCS-3B
Year: 2025 Fall

UML Diagrams

Use Case Diagram



Class: BSCS-3B
Year: 2025 Fall

Tools and Technologies

- **Language:** C++
- **Compiler:** Dev-C++ / Visual Studio
- **Libraries:** *<iostream>*, *<conio.h>*, *<windows.h>*, *<fstream>*, *<vector>*, *<string>*, *<algorithm>*

Features

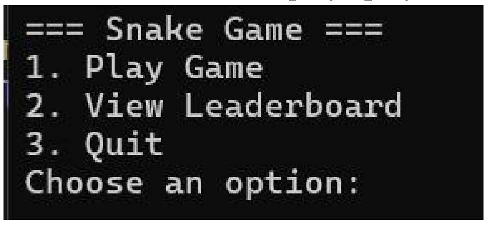
1. **Menu System:** Play / View Leaderboard / Exit options.
2. **Dynamic Snake Body:** Implemented using doubly linked list nodes.
3. **Real-time Movement:** Controlled via WASD or arrow keys.
4. **Collision Detection:** Ends game upon wall or self-collision.
5. **Leaderboard:** Stores top 10 player scores.

Implemented Concepts

- **Doubly Linked List** for snake structure.
- **OOP Design** through encapsulated classes.
- **File Handling** for persistent leaderboard.
- **STL Algorithms** (*sort*) for ranking scores.

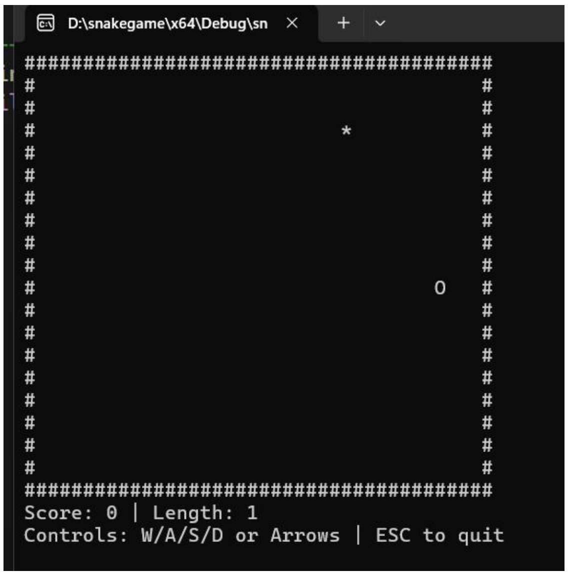
Output (Example Screens)

1. **Main Menu** – Displays play and leaderboard options.



```
=== Snake Game ===
1. Play Game
2. View Leaderboard
3. Quit
Choose an option:
```

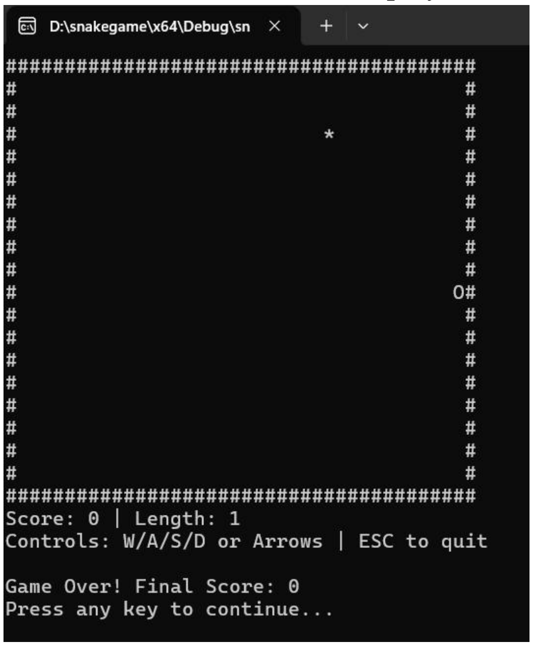
2. **Gameplay Screen** – Snake, food, and border rendering.



D:\snakegame\x64\Debug\sn

```
########################################
#                                      #
#                                      #
#                          *           #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                  O   #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
########################################
Score: 0 | Length: 1
Controls: W/A/S/D or Arrows | ESC to quit
```

3. **Game Over Screen** – Displays final score.

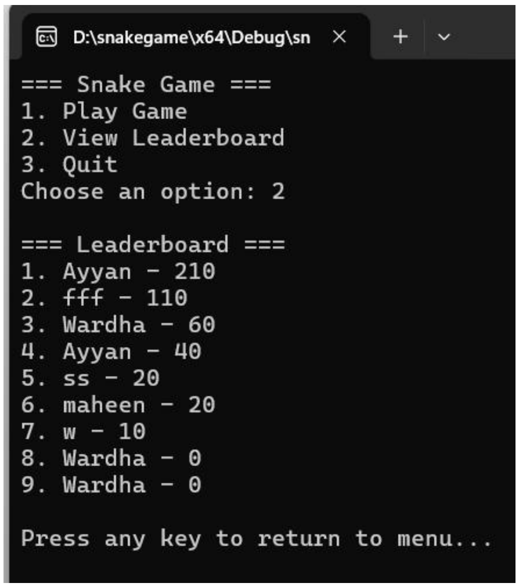


D:\snakegame\x64\Debug\sn

```
########################################
#                                      #
#                                      #
#                          *           #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                     O#
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
#                                      #
########################################
Score: 0 | Length: 1
Controls: W/A/S/D or Arrows | ESC to quit

Game Over! Final Score: 0
Press any key to continue...
```

4. **Leaderboard Screen** – Sorted player scores.

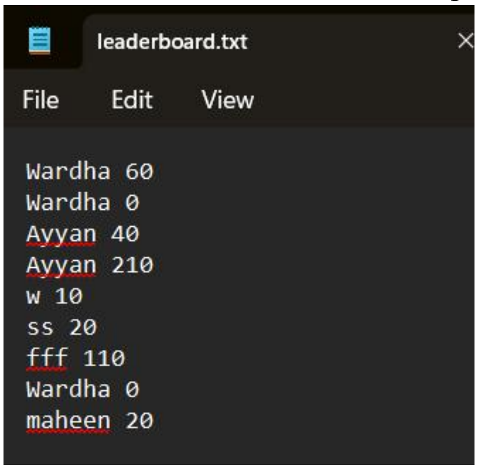


```
=== Snake Game ===
1. Play Game
2. View Leaderboard
3. Quit
Choose an option: 2

=== Leaderboard ===
1. Ayyan - 210
2. fff - 110
3. Wardha - 60
4. Ayyan - 40
5. ss - 20
6. maheen - 20
7. w - 10
8. Wardha - 0
9. Wardha - 0

Press any key to return to menu...
```

5. **Leaderboard txt file** – Sorted player scores and names.



```
Wardha 60
Wardha 0
Ayyan 40
Ayyan 210
w 10
ss 20
fff 110
Wardha 0
maheen 20
```

Conclusion

The project successfully integrates major C++ programming concepts to create a fully functional Snake Game. It demonstrates **dynamic memory usage**, **OOP design**, and **file management** in a practical setting. Future improvements can include GUI implementation using SFML or SDL and level-based gameplay.

References

1. Stroustrup, B. *The C++ Programming Language*, 4th ed. Addison-Wesley, 2013.

2. Microsoft Learn: *Windows Console Functions*

3. GeeksforGeeks: *Linked List Data Structure*

4. C++ Reference: *File Streams*