Course Code: CSE 115 section:2 | Group: 4

Members:

1. Fardin Hossain (ID: 2512532642)

2. Jayonti Sarkar (ID: 1911069042)

3. Afif Chowdhury (ID: 2513880642)

4. Mohammad Ali (ID: 2512818642)

5. Md Nahim (ID: 2514251042)

Abstract

This report provides an update on the progress of the Snake game project, highlighting key developments in game initialization, input handling, snake mechanics, rendering systems, and food and score management. The project aims to enhance programming skills in game development while applying principles of object-oriented design and efficient algorithms.

1. Introduction

The Snake game project serves as a practical application of various programming concepts, including game design, user input processing, and dynamic rendering. This update outlines the contributions of each team member and the current status of the project.

2.Project Overview:

We developed a terminal-based Snake game in C using a modular design Each team member implemented a critical component, ensuring efficient collaboration.

The Snake game incorporates the following features: Smooth snake movement, input handling, Food collection and score tracking, Collision detection with walls and the snake itself, Stable terminal rendering.

3. Team Contributions

3.1 Game Initialization (Fardin Hossain)

Responsibility: Setup game state and control flow.

Key Features: Centralized game state management

Randomized seed initialization via srand(time(0))

Loop timing with Sleep(200) for smooth updates

3.2 Input Handling(Jayonti Sarker)

Responsibility: The input handling module is designed to detect keyboard inputs non-blockingly, allowing for real-time movement control of the snake. The Input() function utilizes _kbhit() to check for key presses and _getch() to read the input. The directional control is managed as follows:

Pressing **'a'** changes the direction to **LEFT**, unless the current direction is **RIGHT**. Pressing **'d'** changes the direction to **RIGHT**, unless the current direction is **LEFT**. Pressing **'w'** changes the direction to **UP**, unless the current direction is **DOWN**. Pressing **'s'** changes the direction to **DOWN**, unless the current direction is **UP**. Pressing **'x'** sets the gameOver flag to **1**, terminating the game.

Key Features: Direction validation to prevent 180° turns

Instant exit with 'X' key

Responsive input polling using conio.h

3.3 Snake Mechanics (Afif Chowdhury)

Responsibility: Move snake and detect collisions.

Key Features: O(n) tail movement algorithm

Boundary checks for walls
Self-collision detection loops.

3.4. Rendering System (Mohammad Ali)

Responsibility: Display game visuals.

Key Features: Flicker-free updates via cursor repositioning

Dynamic borders using ASCII characters

Real-time score display

3.5. Food & Score (Md Nahim)

Responsibility: Spawn food and update score

Key Features: Conflict-free food spawning (avoids snake body)

Score increment system with simple API

Future Work & Next Steps

- **Bug Fixes:** Improve input handling and movement logic.
- **Graphics & UI:** Add animations, start menu, and game-over screen.
- **New Features:** Implement difficulty levels, pause/resume, and sound effects.
- **Code Improvement:** Refactor code and add documentation.
- **Final Testing:** Debug and optimize performance for a smoother experience.

7. Conclusion

As of this update, the project is progressing well, with each team member making significant contributions to their respective areas. The next steps involve refining game mechanics, conducting testing, and preparing for the final report.