Course Code: CSE115 | Section: 04

Project Group No.: 4

Project Name: Snake Game – Project Update Report

Faculty Initial's: MSRB

Group Members:

1. Farhan Haque Labib - 2524053642
2. Raonok Matabber – 2523862642
3. Tahsin Rahman - 2523961642
4. Md Tanvir Hossain Sisir - 2522060642
5. Mehebub Hassan Sorno - 2523427642

Abstract

The report focuses on compiling an overview of initial design and development progress for our Snake Game project implemented in C programming language. The game logic demonstrated real-time input operation with basic console graphics, wall collision detection, and a very basic score system. The project reinforces our understanding of modularized code, structured programming, and development together through GitHub workflows. This update contains our work so far, technical challenges faced, and clear-cut plans toward completing the remaining features within the semester timeline.

1. Introduction

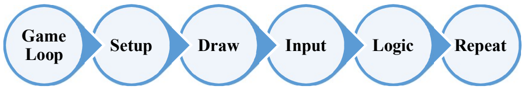
The Snake Game is a classic arcade game which is always recreated in different programming languages to help beginner coders understand the structure and concepts of that programming language. It has a very simple but slightly complex logic which helps coders to practice loops, conditionals and data structures in a real-time project. By developing this project on C, we have understood how to work with enums, structs and arrays. Our version uses standard libraries together with `<windows.h>` and `<conio.h>` for screen refresh and non-blocking keyboard input. While working we understood the importance of great teamwork, ideas, version, revision practices which will help us in the long road of software engineering.

2. Methodology

The project follows a modular coding approach for clarity and maintainability.

- **Structs:** We use custom data structures such as position and gamestate to manage the snake's head location, body array, length, direction, score, and overall game status.

- **Modules:** Separate functions — `setup()`, `draw()`, `input()`, and `logic()` — keep the program logic organized and easy to debug.



Game Loop → Setup → Draw → Input → Logic → Repeat

- **Real-Time Input:** We handle user input using `_kbhit()` and `_getch()` to detect key presses instantly without blocking the main loop, enabling smooth movement in all directions.

- **Console Graphics:** The game area is rendered using simple ASCII characters. We clear and redraw the console using `system(\"cls\")` for each frame, ensuring the snake's movement is visible in real-time.

- **GitHub Version Control**: Our group uses a public GitHub repository to track progress, manage code changes, and share files. Regular commits and updates ensure that every group member can collaborate effectively and resolve any merge conflicts early.

3. Current Progress

DONE Basic snake movement in all four directions (UP, DOWN, LEFT, RIGHT) is functional and responsive.

DONE Wall collision detection works as intended; the game ends if the snake hits any boundary.

IN PROGRESS We are currently implementing the logic for generating food at random positions within the grid.

IN PROGRESS The score system is partially implemented and will be updated to increase whenever the snake eats food. We are also working on the logic for growing the snake's length after food is consumed and detecting self-collisions to end the game if the snake hits its own body.

4. Challenges

Key technical challenges so far:

- Avoiding flicker when redrawing the console, as `system(\"cls\")` can cause noticeable flashing if not handled efficiently.
- Maintaining smooth real-time input without blocking the main game loop, ensuring the snake responds immediately to key presses.
- Correctly updating the body segments when the snake changes direction, so that self-collision detection works reliably.
- Managing corner cases, such as rapid direction changes that can accidentally reverse the snake's movement, is another area we are carefully testing.
- Ensuring the project structure stays clean and modular even as new features are added.

5. Next Steps

In the coming weeks, we plan to:

- Complete the food spawning logic so that new food appears at random, valid grid positions.
- Ensure the snake's body grows properly after eating food, and that the score updates correctly.
- Implement robust self-collision detection and test edge cases thoroughly to prevent logical bugs.
- Polish the codebase by adding meaningful comments and refactoring redundant parts for clarity.
- Prepare a clear, high-quality 1-minute demo video showing the project running smoothly.
- Draft the final 8-page report using the IEEE double-column template and submit all files via GitHub as required.

6. Instructions Followed

This update report is prepared according to the course manual instructions:

- Written in a professional article format (e.g., IEEE double-column).
- Style inspired by published scientific papers (e.g., Nature, Science, IEEE, Wiley).
- Figures, diagrams, and references will be added in the final version.
- Proper citations and no plagiarism — all external sources, videos, or AI tools will be acknowledged if used.

9. Acknowledgments

We thank Mohammad Shifat-E-Rabbi for sir's guidance and support as course instructor. We also acknowledge OpenAI's ChatGPT & Deepseek & Google's Gemini for helping draft and refine this report, and the open-source community for tutorials and examples that improved our C programming and game logic.

10. References

[1] M. Shifat-E-Rabbi, "Course Manual – Theory," North South University, 2025.

[2] OpenAI, "ChatGPT," https://chat.openai.com, accessed July 2025.

[3] GeeksforGeeks, "Simple Snake Game in C," Available: https://www.geeksforgeeks.org/simple-snake-game-in-c/, accessed July 2025.

[4] W3Schools, "C Programming Language," https://www.w3schools.com/c/c_intro.php, accessed July 2025.

[5] Microsoft Learn, "System Function (Windows)," Available: https://learn.microsoft.com/en-us/cpp/c-runtime-library/reference/system, accessed July 2025.

[6] Google, "Gemini", https://gemini.google.com, accessed July 2025

[7] GitHub, "Snake Game Group Repository," [Online]. Available: https://github.com/FarhanHaqLabib/Snake_Game, Accessed July 2025.