North South University

CSE 115.4 Project

SNAKE GAME

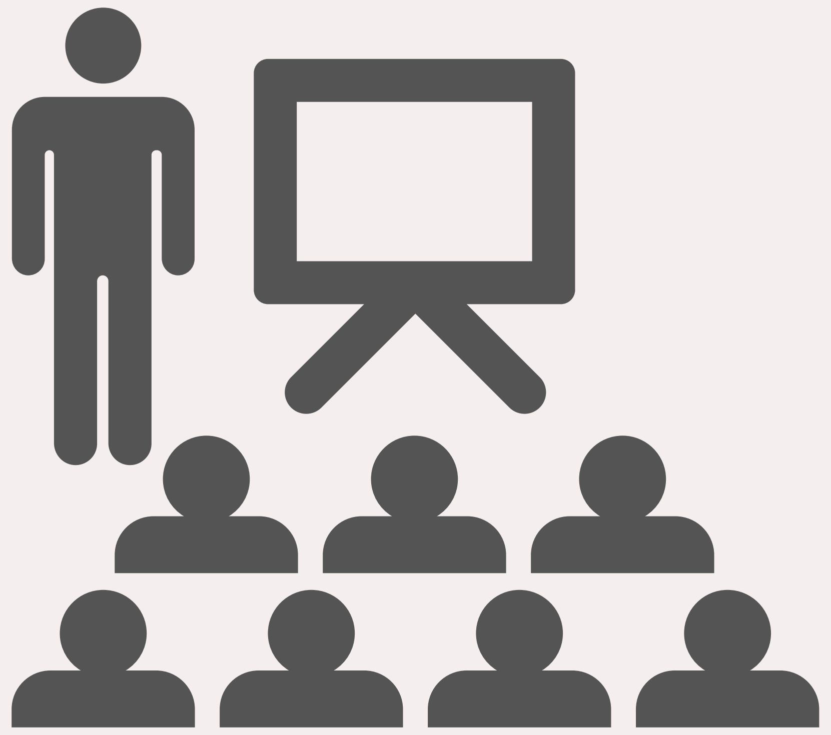
With C Programming

FARABI AHMED | ARGHYA ROY | A K M FAIRUZ KABIR | IMTIAZ AHMED

North South University

INTRODUCTION

We developed a Snake Game as a console application using the C language. Our objective was to replicate the logic of the classic Snake game. We placed a strong emphasis on interactive gameplay and ensuring real time responsiveness within a simple environment.



A K M FAIRUZ KABIR

GOAL

The game takes place inside a bounded grid where the player directs the snake using keyboard input. The main objective is to navigate to food items that appear at random positions on the board. Consuming these items causes the snake to grow longer and increases the player's score. The game runs continuously until a 'Game Over' condition is met, which happens if the snake either collides with the boundary walls or crashes into its own body.

North South University

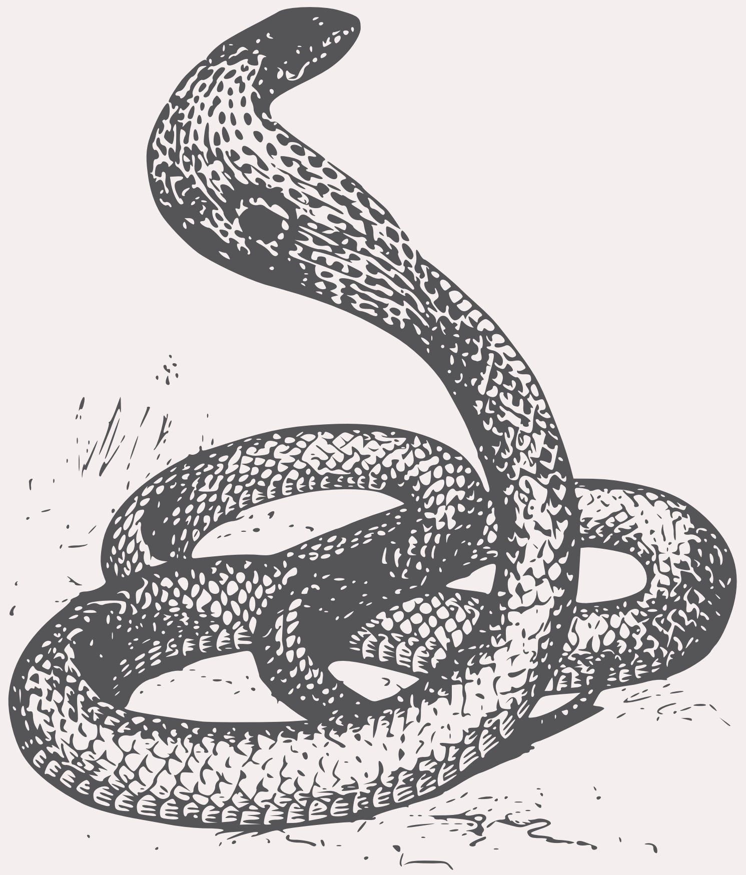
BUILDING THE FOUNDATION

We prioritized clarity and playability by implementing several key features. Visually, the game board is centered on the screen with boundary walls, using distinct symbols: an '@' for the head, asterisks for the body, and a '+' for food, so the player can easily distinguish game elements. The gameplay is driven by a scoring system where every food item consumed awards 100 points, and we included a high-score tracker that remembers your best performance across multiple rounds. To make the user experience seamless, we added simple keyboard commands, such as pressing 'R' allows you to instantly restart the game, while pressing 'Esc' lets you exit at any time.

ARGHYA ROY

North South University

TECHNICAL ANALYSIS



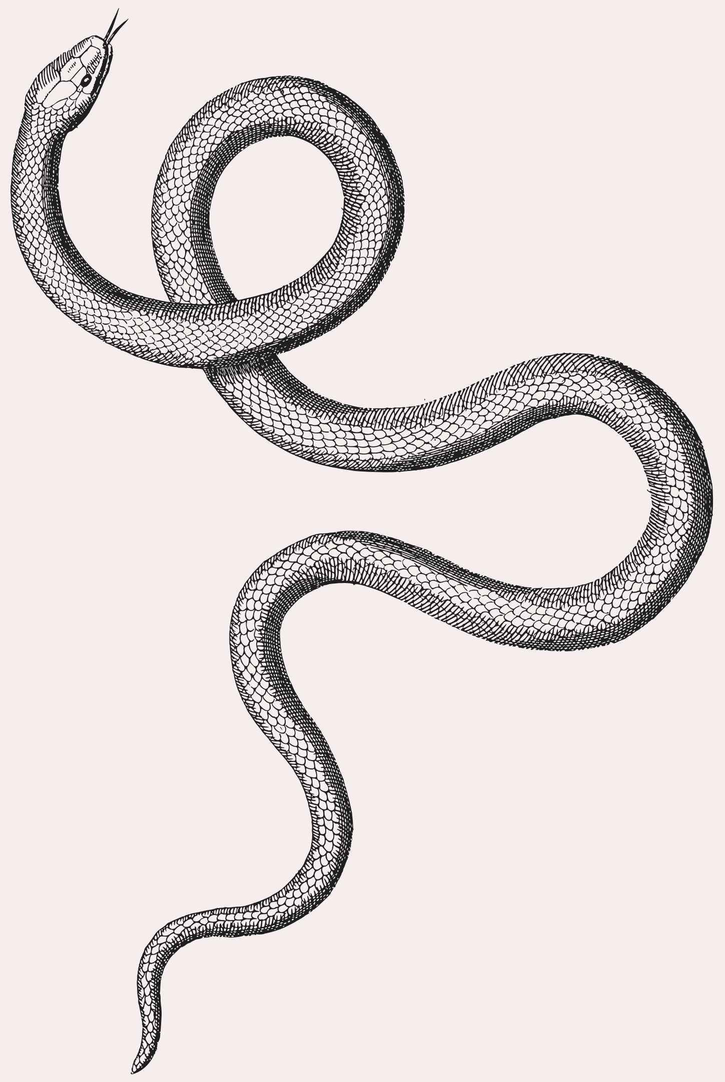
we applied the core logic learned in class by using Structures to efficiently manage the snake and food objects, and Arrays to map out the game board and track segments. We maintained a clean, modular design by breaking the code into 10 specific Functions—such as draw_snake and move_snake—running within a continuous game loop. Additionally, we utilized key libraries like <conio.h> to handle real-time keyboard inputs instantly without pausing, and <time.h> to seed random number generation, ensuring the food spawns in a different spot for every session.

ARGHYA ROY

North South University

USER GAMEPLAY & DESIGN

We designed the gameplay to be intuitive by utilizing the standard W, A, S, D keys for movement, a setup most gamers are already familiar with. To ensure a smooth experience, the game handles input immediately without pausing and uses a screen-clearing command to refresh the board every frame, creating the illusion of fluid animation. We also focused on the visual layout by applying horizontal and vertical padding to center the board on the screen, ensuring it looks neat rather than being confined to the corner, all while maintaining a balanced difficulty through the dynamic growth of game objects.



ARGHYA ROY

CONCLUSION

To conclude, this project provided a practical way to apply fundamental C concepts by combining loops, arrays, and structures into a well-organized, interactive console application. It demonstrates how to handle complex logic such as collision detection and real time input, while maintaining a modular design that leaves room for future improvements like sound effects or difficulty levels.

North South University
CSE 115.4 Project

THANK YOU

FARABI AHMED | ARGHYA ROY | A K M FAIRUZ KABIR | IMTIAZ AHMED