­

**PROJECT REPORT**



Classic Snake Game

**TEAM MEMBERS:**

* Aly Rahim (k24-0512) -> Leader
* Tooba Arshad (k24-0658)
* Ayesha Faisal (k24-0751)

BCS (1-E)

**SUBMITTED TO:**

MS. KHADIJA TUL QUBRA

**INTRODUCTION**

Our project is a classic Snake Game that offers users the ability to read game instructions, view the top 10 high scorers, and enjoy smooth, engaging gameplay. It’s designed to ensure a seamless and enjoyable experience for players.

Snake is a straightforward game with a simple structure that aligns well with the capabilities of C, making it ideal for beginners or those practicing game development. The Snake Game primarily relies on logical concepts like movement, collision detection, and score calculation, which can be effectively implemented using C's structured programming approach.

**DESIGN AND IMPLEMENTATION**

1. **Main Menu:**

* Display options to the player:
  1. Start Game
  2. View Top 10 Leaderboard
  3. View Instructions
  4. Quit Game
* Use arrow keys or other inputs to navigate options.
* On selection:
  1. Start Game calls the *Game Loop.*
  2. Leaderboard displays the top scores from a file.
  3. Instructions explain the game rules and controls.
  4. Quit exits the program.

2. **Game Initialization:**

* Reset the snake position, length, score, and game state variables.
* Randomly generate the initial position of the food.
* Clear the screen to prepare for gameplay.

3. **Game Loop (Play):**

* Continuously update the game state until the game is over:
  + *Draw Game***:** Display the walls, snake, food, and score on the console.
  + *Input Handling:* Capture real-time player input for movement (W/A/S/D).
  + *Game Logic:*
    - Move the snake based on input.
    - Check for collisions with walls, the snake's tail, or food.
    - Update score and snake length if food is eaten.
    - End the game if a collision occurs.
  + Add a small delay (using Sleep) to control the game's speed.

4. **Scoring and High Scores:**

* Maintain a leaderboard of the top 10 scores.
* After game over:
  + Prompt the player to enter their name.
  + Check if the score qualifies for the top 10.
  + If so, update the leaderboard with the player's score.
* Save the updated leaderboard to a file for future sessions.

5. **Game Over and Reset:**

* Display the "Game Over" screen with the player's final score.
* Allow the player to return to the main menu or exit.

6. **File Management for High Scores:**

* *Load High Scores:*
  + On program start, read scores from a file into memory.
* *Save High Scores:*
  + After each game, write updated scores back to the file.

**1. Main Program Flowchart:**

Load Highscores from File

Display Main Menu

No

Yes

Player Selection?

Other Options

Start Game

Instructions/Leaderboard

Initialize Game

Return to Menu

Enter Game Loop

Game Over: Update Highscores

Save Highscores to File

**2. Main Menu Module Flowchart:**

Display Main Menu

No

Yes

If Player Input == Start Game

No

Yes

Player Input == View Leaderboard

Start Game

Call Game

View Leaderboard

View Instructions

Display Top 10

**3. Game Initialization Flowchart**

Initialize Game Variables

Set Snake Position

Generate Random Food Position

Clear Screen For Game Play

Start Game Loop

**3. Game Loop Flowchart**

Draw Game Screen

Capture Player Input

Update Snake Movement

No

yes

Collision Detected?

Yes

Food Eaten?

Game Over

Increase Snake Length

Generate New Food

**5. Collision Handling Flowchart**

Check Wall Collision

Check Snake Tail Collision

Collision Detected?

No

Yes

Continue

**6. High Score Module Flowchart**

Open File For Reading

Read Scores Into Memory

Close File

Return To Main Page

**Updating High Scores:**

No

Yes

Compare New Score With Leaderboard

Do Nothing

Insert New Score

Shift Lower Scores Down

Update Leaderboard

**Saving Highscores:**

Open File For Writing

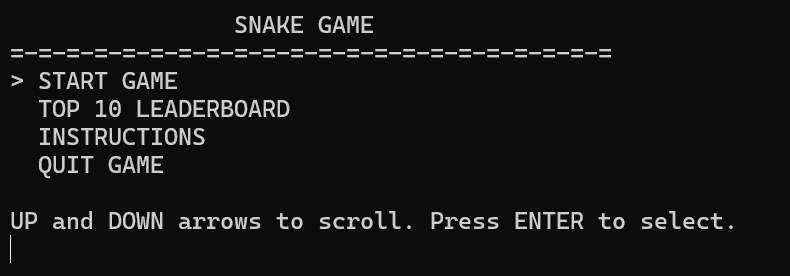
Write Updated Score To File

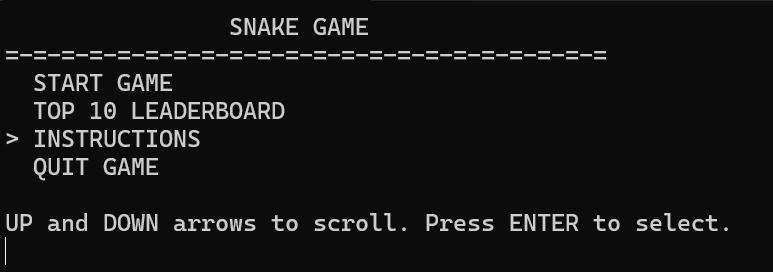
Close File

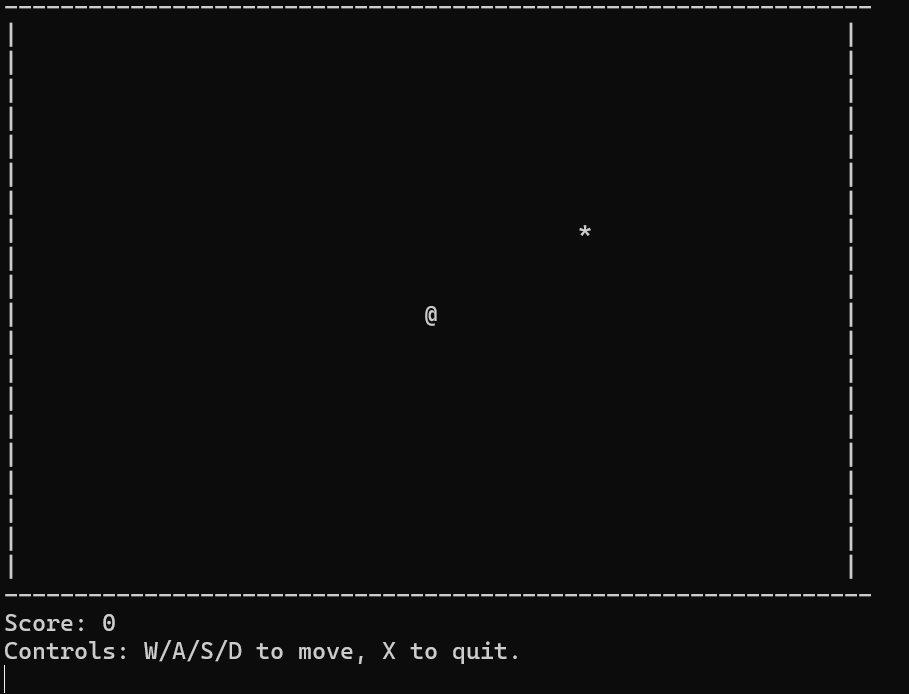
Return To Main Page

**RESULTS**

* **Main Menu:**

**** Figure 1: Cursor Up

**** Figure2 : Cursor Down

 Figure 3: On Start

* **Game Over and Reset:**

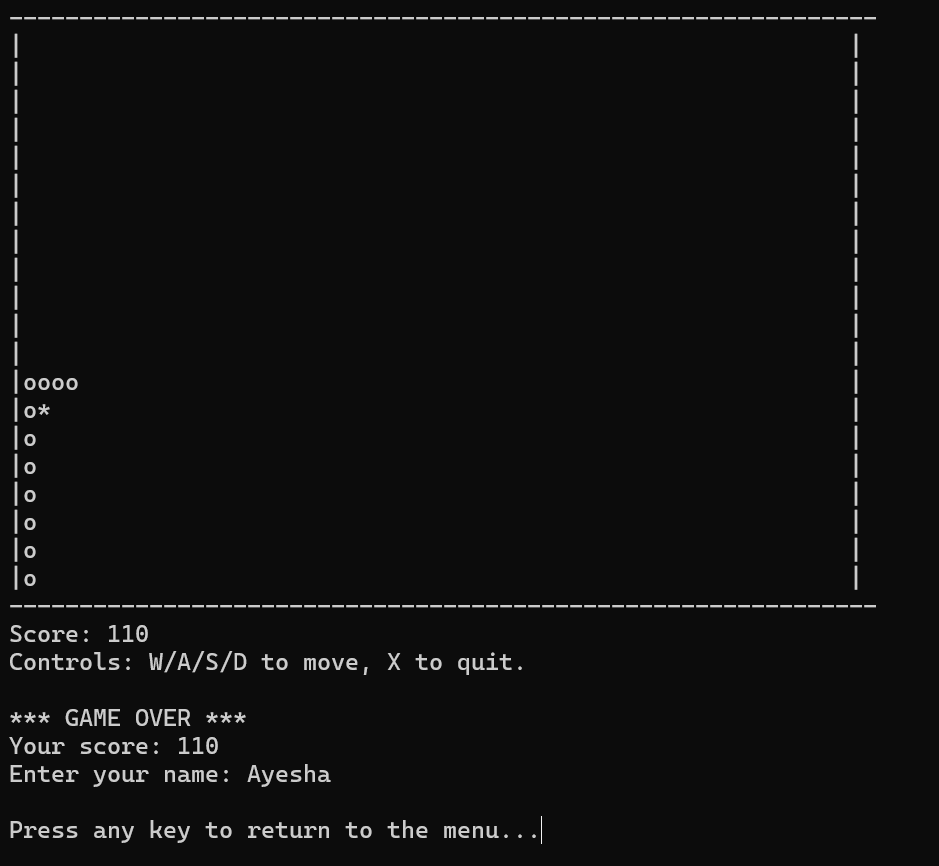
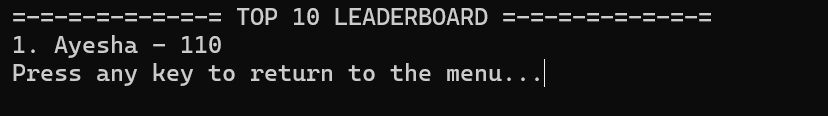


Figure 4: Prompting from user their name to keep the records for high scores

* **File Management for High Scores**



*Figure 5: Leaderboard from menu*

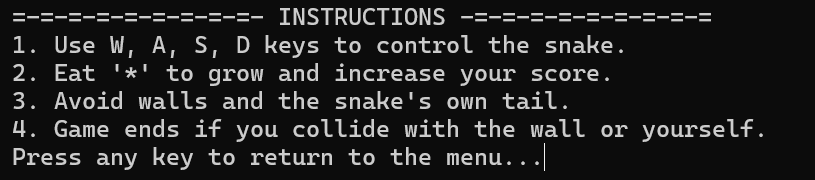


Figure 6: Instructions from menu

**REFERENCES**

* W3 School
* geeksforgeeks
* https://youtu.be/t3y2b2\_moY8?si=07NE7InkLNuF-tH-