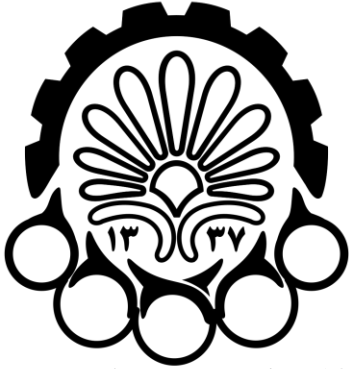


بسمه تعالی



دانشگاه صنعتی امیر کبیر  
( پلی تکنیک تهران )



دانشکده مهندسی کامپیوتر

## Fundamental of Programming Final Project

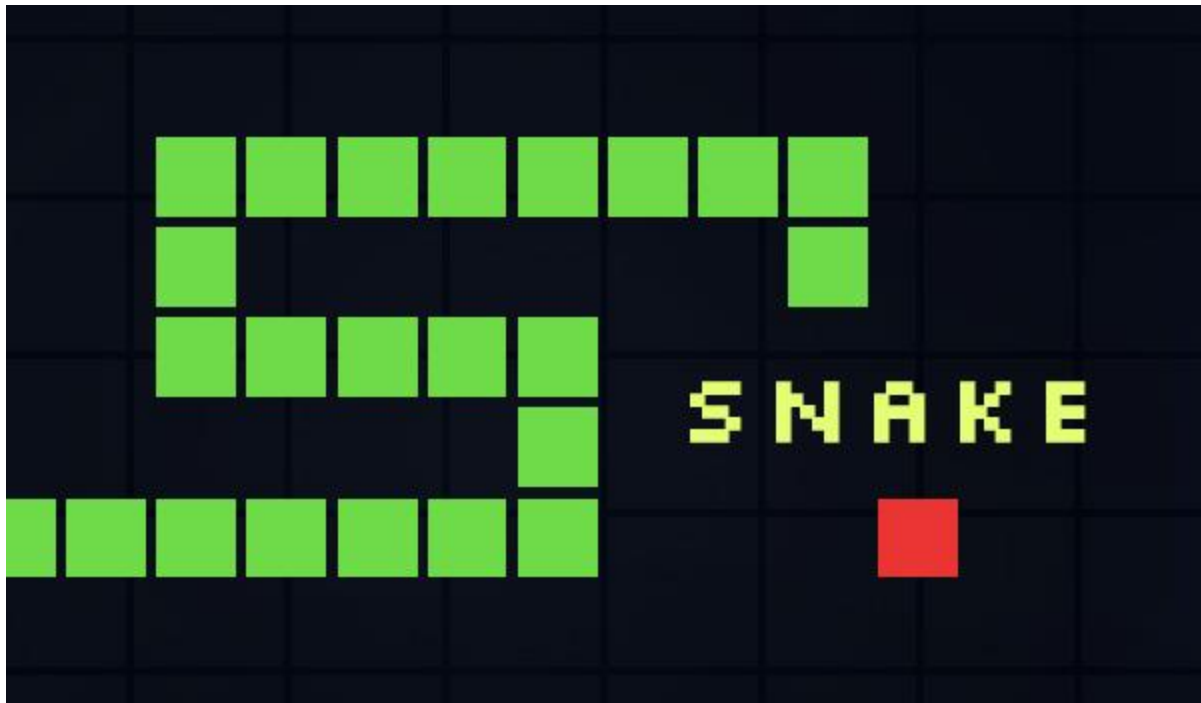
Version 1

Spring 2024

Morteza Zakeri

## Index

Abstract.....	2
Project Breakdown .....	2
1. Setup and Initialization.....	3
2. Game Loop.....	3
3. Snake Movement .....	3
4. Food Generation.....	4
5. Collision Detection .....	4
6. User Input Handling .....	4
7. Graphics and Display .....	5
8. Ending the Game.....	5
Conclusion .....	6



## Abstract

The Snake Game is a classic arcade game that provides an excellent opportunity for students to apply their programming skills in C. This project encompasses essential concepts such as loops, conditionals, arrays, and user input handling. Below is a structured guide to understanding what you need to do to create a Snake Game in C and what the expected outcomes should be.

## Project Breakdown

1. Setup and Initialization
2. Game Loop
3. Snake Movement
4. Food Generation
5. Collision Detection

6. User Input Handling
7. Graphics and Display
8. Ending the Game

## 1. Setup and Initialization

### What to Do:

- Initialize the game by setting up variables for the game state, such as the snake's position, the position of the food, and the score.
- Define the size of the game board.
- Prepare arrays to keep track of the snake's body segments.

### Expected Result:

- A well-defined game state with initial values for the snake's starting position, the initial score set to zero, and the food placed at a random position on the board.

## 2. Game Loop

### What to Do:

- Create a loop that runs continuously until the game is over.
- Within this loop, handle all game updates such as drawing the game state, processing user input, and updating the game logic.

### Expected Result:

- A continuously running game loop that repeatedly updates the game's display, processes inputs, and checks game conditions.

## 3. Snake Movement

### What to Do:

- Implement logic to move the snake in the direction specified by the player's input.
- Ensure the snake's body follows its head, creating a slithering effect.

**Expected Result:**

- The snake should move in the correct direction each frame, and its body should follow smoothly behind its head.

**4. Food Generation****What to Do:**

- Place a piece of food at a random location on the board.
- Ensure that the food is not placed on the snake's body.

**Expected Result:**

- A piece of food appears at a random position on the board. When the snake eats it, a new piece of food should appear at another random position.

**5. Collision Detection****What to Do:**

- Detect collisions between the snake's head and the walls or its own body.
- If a collision occurs, set the game state to over.

**Expected Result:**

- The game should end if the snake runs into the walls or its own body, indicating a game-over condition.

**6. User Input Handling****What to Do:**

- Read user inputs to change the snake's direction using keys such as W (up), A (left), S (down), and D (right).
- Ensure the snake cannot immediately reverse direction (e.g., from left to right).



**Expected Result:**

- The game ends gracefully, displaying the player's score and a message indicating that the game is over.

## Conclusion

The Snake Game project in C is an engaging way to apply fundamental programming concepts. By following the steps outlined, students will create a fully functioning game that reinforces their understanding of arrays, loops, conditionals, and user input. The result will be a classic, interactive game that is both fun to play and rewarding.