**Summer Training Report**

### on

**Snake Game**

# (Using Data Structures in C++)

## A Project Report submitted in partial fulfillment of the requirements for the award of

**Bachelor of Engineering**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

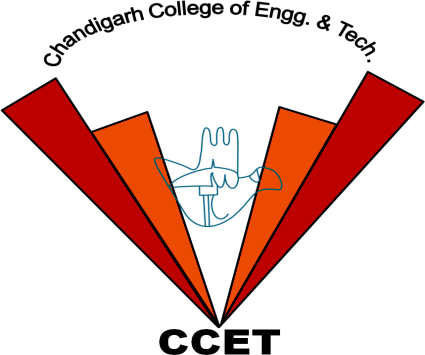
### Submitted by

**Devashish Gupta**

**(Roll no:CO20314)**

**Under the supervision of**

**Dr. Varun Gupta**



**CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(DEGREE WING)**

Government Institute under Chandigarh (UT) Administration, Affiliated to Panjab University

, Chandigarh

Sector-26, Chandigarh. PIN-160019

## August, 2021

**Department of Computer Sc. & Engineering**

## CANDIDATE’S DECLARATION

I hereby declare that the work presented in this report entitled “SNAKE GAME (Using Data Structures in C++”, in fulfillment of the requirement for the award of the degree Bachelor of Engineering in Computer Science & Engineering, submitted in CSE Department, Chandigarh College of Engineering & Technology (Degree wing) affiliated to Punjab University, Chandigarh, is an authentic record of my/our own work carried out during my degree under the guidance of Dr. Varun Gupta. The work reported in this has not been submitted by me for award of any other degree or diploma.

Date: 09-08-2021 Devashish Gupta

Place: Chandigarh CO20314

## Department of Computer Sc. & Engineering

### CERTIFICATE

This is to certify that the Project work entitled “SNAKE GAME (Using Data Structures in C++)” submitted by Devashish Gupta. CO20314 in fulfillment for the requirements of the award of Bachelor of Engineering Degree in Computer Science & Engineering at Chandigarh College of Engineering and Technology (Degree Wing), Chandigarh is an authentic work carried out by him/her under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree.

Date: 07-08-2021 Dr. Varun Gupta

Place: Chandigarh Dept. of CSE

CCET (Degree Wing)

Chandigarh

## Department of Computer Sc. & Engineering

### ACKNOWLEDGEMENT

The credit for the successful completion of this project work goes beyond our own work, to those people who have always been with us throughout. And we take this opportunity to express our heartfelt gratitude to each one of them.

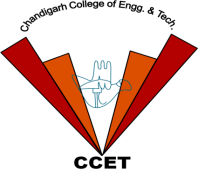
We express our sincere thanks to our Mentor Dr. Varun Gupta of for his valuable suggestions and providing guidance throughout the project, that enabled us to complete this project successfully.

Devashish Gupta

CO19335

(2nd Semester)

Dept. of CSE

**CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY (DEGREE WING)**

Government Institute under Chandigarh (UT) Administration | Affiliated to Panjab University , Chandigarh

Sector-26, Chandigarh. PIN-160019 | Tel. No. 0172-2750947, 2750943 Website: [www.ccet.ac.in](http://www.ccet.ac.in/) | Email: [principal@ccet.ac.in](mailto:principal@ccet.ac.in) | Fax. No**. :**0172-2750872

## Department of Computer Sc. & Engineering

### ABSTRACT

I attempted to recreate the famous Snake game, in which the player controls a snake-like line that grows in length, with the line serving as the primary barrier.

Snake has been around since the late 1970s, and if you owned a Nokia phone, you were definitely familiar with it. The cursor keys are used to manipulate a snake as it moves around the screen. Fruit is required for growth. You are not permitted to contact the exterior walls or your own body.

### CONTENTS

Candidate’s declaration i

Certificate by the guide ii

[Acknowledgement iii](#_TOC_250001)

[Abstract iv](#_TOC_250000)

List of tables v

List of figures vi

Chapter 1 – Introduction to Data Structures 1

1.1 <Name of the topic>………………………………………….

1.2 < >…………………………………………………….. Chapter 2 - <Name of the Chapter> ……………………………………………

### CHAPTER 1: INTRODUCTION TO DATA STRUCTURES

* 1. **Data Structures**

A data structure is a data organization, management, and storage format in computer science that allows for quick access and modification. A data structure is an algebraic structure about data that contains a collection of data values, their relationships, and the functions or operations that may be applied to the data.

For applications such as big databases and internet indexing services, data structures provide a way to efficiently manage massive amounts of data. In most cases, building efficient algorithms necessitates the use of efficient data structures. Data structures, rather than algorithms, are emphasized in some formal design methodologies and programming languages as the fundamental organizing force in software design. Data structures can be used to organize information storage and retrieval in both main and secondary memory.

* 1. **Types of Data Structures**
     1. Arrays

An array is a data structure that represents a grouping of related data components. This signifies that all of the array's items are of the same or homogenous data type. It can be of the type Integer, Character, or Float. The Data Type Name comes first, followed by the Variable Name and its Capacity or Size. For instance, int a [10] creates an array of integer type with a size of 10.

Similarly, we must provide the Variable Name with the index number of the element we wish to access in order to access any element of an array. For example, a [5] returns the member of array an at index 5.

* + 1. Files

A file is a grouping of data or information. It's kept in backup storage devices. We may do a variety of procedures in File as well. We use File to store data in a secondary storage device when the data is substantial enough.

* + 1. Lists

List is a Data Structure that uses Dynamic Memory Allocation to store, retrieve, and perform a variety of operations. Unlike an array, the memory for each element in a list is allocated dynamically. Linear and non-linear lists are the two types of lists. Take a peek at the following Lists.

* + - 1. Linear List
* Linked List

A Linked List is a data structure with numerous nodes that is linear. Each node is made up of a Data Item and a Pointer that points to the next node. The node's pointer variable is used to point to the next node.

* Stack

Stack is a linear Data Structure that is similar to an array in that it has an orderly collection of data pieces. However, unlike an array, we can only enter and retrieve data from one end. Push and Pop are the two operations of entering and retrieving data from Stack. Push operation is used to add a value to the stack, while Pop operation is used to recover or access the item from the stack. It's worth noting that we can only do Push and Pop operations from one end.

* Queue

Queue is a linear Data Structure that is similar to an array, except that we can only enter values from one end and access them from the other. The Rear End is the node from which we enter or add the element, and the Front End is the node from which we can access the element.

* + - 1. Non-linear Lists
* Tree

Tree is a Non-Linear Data Structure that stores its elements in a hierarchical way, as its name suggests. As a Non-Linear Data Structure, it is not necessary to have items in a tree in a sequential order. There will be one Root Node at the top of the Tree, followed by its Child Node, and those Child Nodes can build as many sub-trees as needed.

* Graph

G = {V, E} represents a Graph, which is a Non-Linear Data Structure. Vertices are represented by V, and Edges are represented by E. With the use of Edges, distinct Vertices in a Graph are connected. Various Edges linking the Vertices can be given different weights or costs. If E1 is the edge linking V1 and V2, for example, we can write E1 = {V1, V2}.

### CHAPTER 2: SNAKE GAME

* 1. **Introduction**

The following is an example of a game written in C++ using Data Structures and is based on a game called ‘Snake’ which has been around since earliest days of home computing and has re-emerged in mobile phones during recent years.

* 1. **Playing the Game**

The goal of the game is to collect as many stars (food) as possible while avoiding the obstacles (borders and the snake itself). Use the arrow keys to change the direction in which the snake travels. The snake grows longer as you collect food, increasing your chances of colliding with oneself. As you go on eating food, the speed with which the snake travels increase gradually. The length of the snake and the amount of food eaten determines your score. There is no such thing as a life.

* 1. **Flowchart**

MAIN MENU

Back to Main Menu

Enter your Choice

View Scores

New Game

Enter Name:

YES

NO

Do you want to play again?

Play Game

* 1. **Explanation**
* The project is written in C++ language using classes. The program is based on “linked list” data structure, about which a little explanation is already given in the previous chapter. Different header files, for example, “bits/stdc++.h”, “conio.h”, “windows.h” are used for accessing different pre-defined set of functions.
* As we run the program the 1st window that appears is the Main Menu, which provides us with the option to play a new game or view the previous scores (will be explained later in the report). After choosing the new game option the output displays some set of rules and instructions required to play the game. As we move on the program ask to enter the player’s name after which the game begins.
* For the purpose of moving the cursor to different co-ordinates gotoxy() function is used which is available in windows.h header file. The alignment of text and boxes is also achieved with this function. The food is generated at random co-ordinates using rand () function available in bits/stdc++.h header file. The movement of snake is coordinated with every frame which changes continuously. Whenever the snake eats the food, its length is incremented by 1 unit. Simultaneously, the speed of snake also increases as we move on. Keep in mind one thing the normal speed of snake is completely dependent on your system specs (RAM, processor, screen fps, etc.). If you run this game on a high-end system, it might be possible that the speed of snake becomes uncontrollably fast. In that case you can always increase the sleep timer of in the program which will eventually decrease the speed of the snake.
* The game only ends when either you crash into a wall or yourself. In that case the condition on which the loop was running breaks and you exit the play function. There is no concept of lives in the game so you only have a single chance. You can always pause the game using “p” or “P” key, and resume it by pressing any other key other than the one’s used for pausing it.
* The moment your game ends you name and score gets written in the linked text file, which you can access by the “View Scores” option in the main menu.
* The scoring system is simple you get 100 points each time you eat the food.