IAMR Group of Institutions

Institute of Applied Medicines & Research

Major Project – File

Subject: ***Major Project***

Subject Code: ***BCA – 605***

Submitted To

Mr. Ajay Tyagi

HOD of BCA

Mr. Alekh Chaudhary

Assist. Professor

Submitted By

Student Name: Mohd Shahrukh

Student Roll No: 1709661798

Course/Semester: BCA VIth Sem

Institute of Applied Medicines and Research

Duhai Ghaziabad UP

BCA Department

Major Project

On

“Snake Game”

Submitted by

Mohd Shahrukh

Roll No: 1709661798

CERTIFICATE

This report on "SNAKE GAME" is a bonafide record of the Major-Project work submitted by

MOHD SHAHRUKH
ROLL NO. 1709661798

In the sixth semester/third year of BCA (Bachelor of Computer Application) during the academic year 2017-2020

Guide
Observer HEAD OF DEPARTMENT

Candidate's Declaration

We hereby declare that the work presented in this project titled "SNAKE GAME" submitted towards completion of Major - Project in Sixth semester of BCA at the Institute of Applied Medicines and Research, GZB is an authentic record of our original work pursed under the guidance of Prof. Alekh Chaudhary Sir, IAMR GZB.

We have not submitted the matter embodied in this project for the award of any degree.

MOHD SHAHRUKH

Place: Ghaziabad Date:

ACKNOWLEDGMENT

First and foremost, we would like to express our sincere gratitude to our Major project guides, Prof. Alekh Chaudhary Sir. We were privileged to experience a sub stained enthusiastic and involved interest from their side. This fueled our enthusiasm even further and encouraged us to boldly step in to what was a totally dark and unexplored expanse before us.

ABSTRACT INTRODUCTION

Introduction: Snake game is [1] a type of arcade maze games originally developing from "Blockade" which has been developed by Gremlin Industries and published by Sega in October 1976 [6]. There are many clone games based on Blockade game inspiration, e.g., Bigfoot Bonkers, Surround, Dominos, etc. [2,4]. Snake games are considered to be a skillful game, players try to achieve maximum score as high as possible. We show, in Fig. 1a screenshot of Snake game on Nokia 3310.Nokia is well-known for putting Snake game in their phones [5,13]. Towards the end of the year 2000, Nokia released one of the most successful phones, Nokia 3310, whose Snake game is so popular. Snake games are still included in some new phones from Nokia and available for all platforms. The history of Snake games on Nokia mobile phones [14] is shown in Table 1.We can control the snake in Snake game by using the four direction buttons relative to the direction it is heading in. The snake increases its speed as it gets longer by eating

Table of Contents

Topics:-

Project Background

Snake - Intelligent & Multiplayer Snake Game. It is a remake of traditional snake game with Snake – computer controlled opponent snake and multiplayer functionality. This project aims to bring the fun and simplicity of snake game with some new feature. It will include Snake [1], whose aim will be to challenge the human players. It will also have the multiplayer feature that will allow more than one players to play the game over a network. The traditional snake game does not offer much challenge to it's players. Hence players loose interest in the game after playing it for some time. Offering some sort of challenge and adventure to the players will help increase the addictive power of this game. The multiplayer and Snake features of this game will make this game more challenging and interesting. The simplicity of this game makes it an ideal candidate for a Major project as we can focus on advanced topics like multiplayer functionality and Snake. The strong support for networked application in C programming language and availability of high speed network connection layer in present day computers will allow us to build a very efficient multiplayer version of snake game. Function will be used to create stunning graphics and user interface for the game. The computer controlled snake which acts as the opponent to the human players and has some form of intelligence embedded into it. We will refer to this snake as Snake now onwards in this document.

Short Literature Review

This proposal is organized into several subsections. The objectives of proposing the snake game for Major project is justified by the simplicity of the game. This will allow us to focus on more advanced topics like multiplayer functionality and Snake – computer controlled opponent. Information about some of the existing snake games have been collected from the internet. Lots of variants of traditional snake game exists but none of these game provide the experience of a Console Based games. The introduction of Snake is unique to this game and will offer an adventurous experience to the players. A simple block diagram has been presented here which gives some insight into the structure of the game. This block diagram is for illustration purpose. The final block diagram will be more elaborate and will contain all the components of the game. The scope of this project has been discussed taking into account its commercial value and educational value. The educational value of this project will allow us gain insight of the networking concepts used in multiplayer game. Moreover the Snake functionality will provide us a platform to make further research into the field of “Intelligent Games”. A rough project schedule has been presented to help us complete the project in time. Unit of time used in the schedule is “Weeks” as the final date for submission of Major project has not been finalized.

Problem Statement

The traditional snake game does not offer much challenge to it's players. It is impossible to bring out the best playing skill of a players unless a challenge is offered to them. The players loose interest in the game after playing it for sometimes due to lack of challenge. Hence when people hear about snake game, they quickly form an image of old age arcade game. The absence of powerful graphical resources and weak support for networked application in the programming language is one of the factors that has delayed the development of an interesting and challenging version of snake game.

Objectives

This game aims to change the way people think of traditional snake game. It will offer the experience of commercial multiplayer games to the player retaining the simplicity of traditional snake game. The major objectives of this project are:

1. Create a snake game that will have all the functionality of traditional snake games.

2. Introduce multiplayer functionality in the game that will allow several players to play a game simultaneously. It should give the experience of real time games to the players.

3. Introduce Snake (unique feature of this game) to make the game more challenging and interesting. The movement and action of Snake will be controlled by computer whose aim will be to eat the food before human players capture it.

Existing System Study

A large number of games based on the traditional snake game have been developed across large number of platforms (Linux, windows, etc.) and devices (mobile phones, portable game consoles, etc.). These games have lots of variations in them. Some of these games support multiplayer gaming. However none of these game have the feature of "computer controlled snake - Snake" as opponent to the human players. Hence players loose interest in the game after playing it for sometimes.

Existing games:

● Nibbles - A worm game for GNOME. [http://www.gnome.org/projects/gnome-games/] Supports multiplayer network game. No support for computer controlled opponent snake.

● Original Nokia Phone snake game [http://www.johnjohn.co.uk/html/snake.html] this game was included in early nokia mobile phones. It is plain snake game without any additional feature like multiplayer. There are large number of variants of original snake game available on the internet. Almost all of them have same set of features. Some games provide the feature of multiplayer gaming.

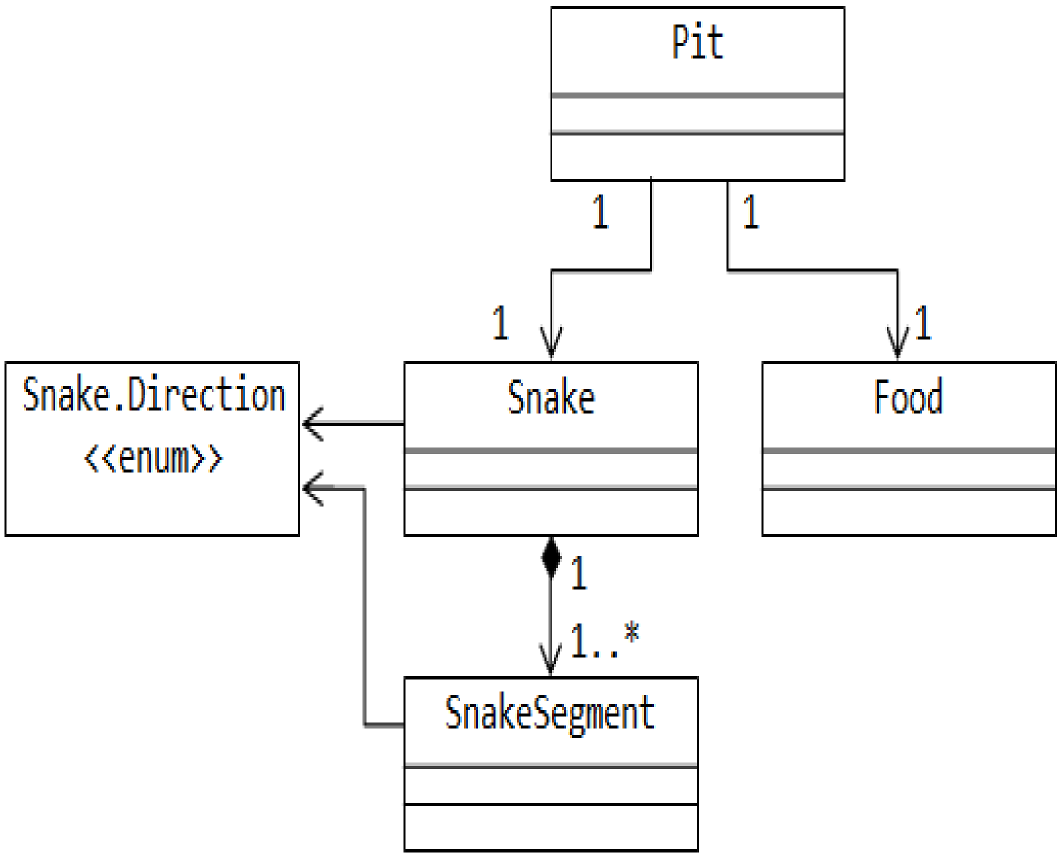
Proposed System

● Description

The proposed game will allow more than one players to play a game simultaneously over a network. The Snake, a feature unique to this game, will offer challenge to the human players and hence make the game more interesting. Functions & Modules will be used to give the user interface (UI) of the game the look of commercial games. The focus of UI will be on usability. As the UI is a very import part of any game, we will follow these UI design guidelines to create attractive game application interface:

1. Simplicity
2. Contrast
3. White Space
4. Balance
5. Alignment

● System Block Diagram



Methodology and Data

● Primary

The several task of this project will be assigned to each team member on the basis of their interest in particular portion of the game. Weekly meeting will be held to assess the status of each member and propose restructuring of plans when required. Project documentation will be performed by the respective team member(s) at the end of each task. Regular discussion with the project mentor will be held to gather suggestions related to the project. It will also provide us chance to update our mentor about the progress of the project.

● Secondary

C-FREE 5 IDE will be the development tool that will be extensively used by all the team members. Several “Developer Collaboration” features of the IDE will be utilized to communicate and collaborate code between the members.

MOTIVATION

Snake is one of most common and popular game of childhood. Its coding is not much difficult, however it will require considerable knowledge on applets & graphics. This makes the project challenging yet approachable. Thus we decided to take up this as our project topic. I started on the snake game when I was taking a break from the maze game. For the snake game I'm trying to use Structured Programming. I stopped this and went back to finish the maze game when I hit a road block. The road block was I am using a function for snake with structure head and nodes. Another function for map. I have to make the snake move around the map and I need to figure out how these to separate function will talk to each other or else I'll have to set it up another way. These separate function are another way I a trying to challenge myself which I think is good as long as I am learning and progressing at a reasonable pace.

Snake Game + Fun = Learning

SNAKE game make learning SNAKE fun. When we're having fun, we're more open to learning. When we're having fun, we wants to keep doing whatever we're doing.

So if you're a parent, teacher, or student and want to make SNAKE Game more user-friendly, come and explore our site where you'll find lots of fun SNAKE activities and other cool SNAKE stuff!

Relevance

<u>Functions used in Snake Game Project in C:</u>

Many functions have been used in this Snake Major project. Here, I will just list them below and describe the functions “gotoxy”, “GotoXY” and “delay” as they are some of the most important functions used in this and many mini projects in C.

void record()
void load()
void Delay(long double)
void Move()
void Food()
void Print()
void Bend()
int Score()
void Boarder()
void Down()
void Left()
void Up()
void Right()
void ExitGame()

void gotoxy (int x, int y) – You need to understand this function as it is one of the most important one used in Snake Game mini project in C. This function allows you to print text in any place of screen. Using this function in Code::Blocks requires coding, but it can be directly used in Turbo C. Here is a code for this function in Code::Blocks.

```
COORD coord = {0, 0};  // sets coordinates to
(0,0) as global variables
void gotoxy (int x, int y)
{
        coord.X = x; coord.Y = y; // X and Y are
the coordinates
        SetConsoleCursorPosition(GetStdHandle(
STD_OUTPUT_HANDLE), coord);
}
```

Here, *COORD coord= {0,0};* is a global variable. It sets the center of axis to the top left corner of the screen.

void GotoXY (int x, int y) – Here is the code for this function in Code::Blocks.

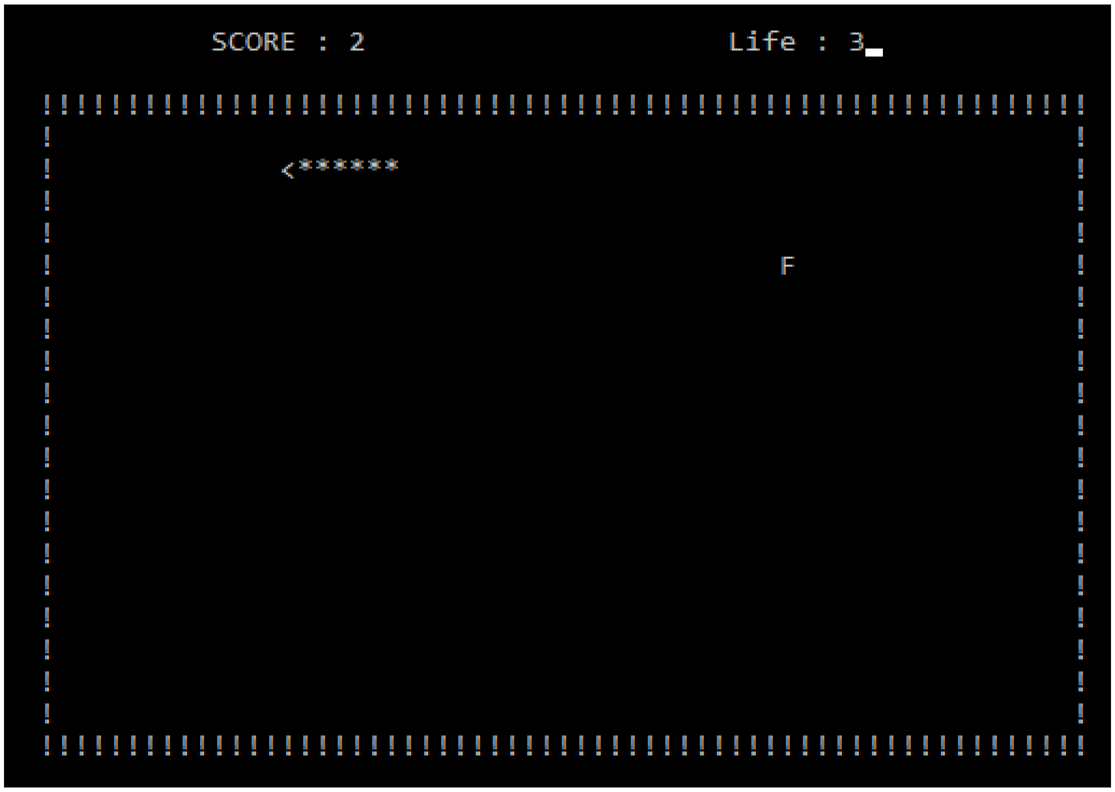
```
void GotoXY(int x, int y)
{
   HANDLE a;
   COORD b;
   fflush(stdout);
   b.X = x;
   b.Y = y;
   a = GetStdHandle(STD_OUTPUT_HANDLE);
   SetConsoleCursorPosition(a,b);
}
```

void delay(long double) – This function delays the execution. It can be used directly in Turbo C, but requires coding in Code::Blocks. The code is given below:

```
void Delay(long double k)

{
   Score();
   long double i;
```

```
    for(i=0; i<=(10000000); i++);
}
```

This Major project in C Snake game gives users a total of three lives to play the game. The life-count decreases as the snake hits the wall or its own body. In this mini project, you can even pause the game in the middle by pressing any key, and you can press any key again to continue.



```
          SCORE : 2                     Life : 3

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                                            !
!             <******                                        !
!                                                            !
!                                                            !
!                                           F                !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!                                                            !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

BASIC CONCEPTS & TOOLS

INTRODUCTION

C is a general-purpose high level language that was originally developed by Dennis Ritchie for the Unix operating system. It was first implemented on the Digital Equipment Corporation PDP-11 computer in 1972.
The Unix operating system and virtually all Unix applications are written in the C language. C has now become a widely used professional language for various reasons.

- Easy to learn
- Structured language
- It produces efficient programs.
- It can handle low-level activities.
- It can be compiled on a variety of computers

Facts about C

- C was invented to write an operating system called UNIX.
- C is a successor of B language which was introduced around 1970
- The language was formalized in 1988 by the American National Standard Institute (ANSI).
- By 1973 UNIX OS almost totally written in C.
- Today C is the most widely used System Programming Language.
- Most of the state of the art software have been implemented using C

Why to use C ?

C was initially used for system development work, in particular the programs that make-up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as code written in assembly language. Some examples of the use of C might be:

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Data Bases
- Language Interpreters
- Utilities

C Compilers

When you write any program in C language then to run that program you need to compile that program using a C Compiler which converts your program into a language understandable by a computer. This is called machine language (ie. binary format). So before proceeding, make sure you have C Compiler available at your computer. It comes along with all flavors of Unix and Linux.

If you are working over Unix or Linux then you can type *gcc -v* or *cc -v* and check the result. You can ask your system administrator or you can take help from anyone to identify an available C Compiler at your computer.

If you don't have C compiler installed at your computer then you can use below given link to download a GNU C Compiler and use it.

C Program Structure

A C Program basically has the following form:-

- Preprocessor Commands
- Functions
- Variables
- Statements & Expressions
- Comments

The following program is written in the C programming language. Open a text file ***hello.c*** using C-Free5 editor and put the following lines inside that file.

A simple hello world program written in C language using C-Free5 IDE.

```
#include <stdio.h>
int main()
{
   /* My first program */
   printf("Hello, World! \n");

   return 0;
}
```

Preprocessor Commands: These commands tells the compiler to do preprocessing before doing actual compilation. Like *#include <stdio.h>* is a preprocessor command which tells a C compiler to include stdio.h file before going to actual compilation. You will learn more about C Preprocessors in C Preprocessors session.

Functions: are main building blocks of any C Program. Every C Program will have one or more functions and there is one mandatory function which is called *main()* function. This function is prefixed with keyword *int* which means this function returns an integer value when it exits. This integer value is returned using *return* statement.

The C Programming language provides a set of built-in functions. In the above example *printf()* is a C built-in function which is used to print anything on the screen. Check Built-in function section for more detail.

You will learn how to write your own functions and use them in Using Function session.

Variables: are used to hold numbers, strings and complex data for manipulation. You will learn in detail about variables in C Variable Types.

Statements & Expressions : Expressions combine variables and constants to create new values. Statements are expressions, assignments, function calls, or control flow statements which make up C programs.

Comments: are used to give additional useful information inside a C Program. All the comments will be put inside /*...*/ as given in the example above. A comment can span through multiple lines.

Note the followings

- C is a case sensitive programming language. It means in C *printf* and *Printf* will have different meanings.
- C has a free-form line structure. End of each C statement must be marked with a semicolon.

- Multiple statements can be one the same line.
- White Spaces (i.e. tab space and space bar) are ignored.
- Statements can continue over multiple lines.

Applications of C

C Programming is a best-known programming language. C Programming is near to machine as well as human so it is called as Middle-level Programming Language. C Programming can be used to do a verity of tasks such as networking related, OS related.

C language is used for creating computer applications. It is used in writing embedded software, Firmware for various electronics, industrial and communications products which use micro-controllers. It is also used in developing verification software, test code, simulators etc. for various applications and hardware products. For Creating Compiles of different Languages which can take input from other language and convert it

Into a lower level machine dependent language. C is used to implement different Operating System Operations. UNIX kernel is completely developed in C Language.

C was initially used for system development work, in particular, the programs that make-up the operating system. Why use C? Mainly because it produces code that runs nearly as fast as code written in assembly language. Some examples of the use of C might be:

1. Operating Systems
2. Language Compilers
3. Assemblers
4. Text Editors
5. Print Spoolers

6. Network Drivers
7. Modern Programs
8. Data Bases
9. Language Interpreters
10. Utilities

In recent years C has been used as a general-purpose language because of its popularity with programmers. It is not the world's easiest language to learn and you will certainly benefit if you are not learning C as your first programming language! C is trendy (I nearly said sexy) - many well-established programmers are switching to C for all sorts of reasons, but mainly because of the portability that writing standard C programs can offer.

Advantages and Limitations of C Language

First of all, let us discuss *what makes* C language the mother of all languages. There are various benefits of C programming that depends on these positive points which can surely define the functionality of C in a better manner.

1. Advantages of C Programming Language

1.1. Building block for many other programming languages

C is considered to be the most fundamental language that needs to be studied if you are beginning with any programming language. Many programming languages such as Python, C++, Java, etc. are built with the base of the C language.

1.2. Powerful and efficient language

C is a robust language as it contains many data types and operators to give you a vast platform to perform all kinds of operations.

1.3. Portable language

C is very flexible, or we can say machine independent that helps you to run your code on any machine without making any change or just a few changes in the code.

1.4. Built-in functions

There are only 32 keywords in ANSI C, having many built-in functions. These functions are helpful when building a program in C.

1.5. Quality to extend itself

Another crucial ability of C is to extend itself. We have already studied that the C language has its own set of ***functions in the C*** library. So, it becomes easy to use these functions. We can add our own functions to the C Standard Library and make code simpler.

1.6. Structured programming language

C is structure-based. It means that the issues or complex problems are divided into smaller blocks or functions. This modular structure helps in easier and simpler testing and maintenance.

1.7. Middle-level language

C is a middle-level programming language that means it supports high-level programming as well as low-level programming. It supports the use of kernels and drivers in low-level programming and also supports system software applications in the high-level programming language.

1.8. Implementation of algorithms and data structures

The use of algorithms and data structures in C has made program computations very fast and smooth. Thus, the C language can be used in complex calculations and operations such as MATLAB.

1.9. Procedural programming language

C follows a proper procedure for its functions and subroutines. As it uses procedural programming, it becomes easier for C to identify code structure and to solve any problem in a specific series of code. In procedural programming ***C variables*** and functions are declared before use.

1.10. Dynamic memory allocation

C provides dynamic memory allocation that means you are free to allocate memory at run time. For example, if you don’t know how much memory is required by objects in your program, you can still run a program in C and assign the memory at the same time.

1.11. System-programming

C follows a system based programming system. It means the programming is done for the hardware devices.

So, with this, we are aware of *why C considered a very powerful language and* why *is it important to know the advantages of C?*

When we study anything new, it becomes important to know the benefits that we gain from that technology. This allows us to grow our interest and implement our knowledge in a practical scenario. Now, let us move on to the “Advantages and Limitations of the C Programming Language”.

2. Limitations of C Programming language

We have already discussed the advantages of C.

You might be thinking about why we are not approaching the language practically and studying the theoretical part in every tutorial. It is because if you will understand the basic functionalities of the language and the methods or operation of the programming language, it becomes easy for you to know whether this language is *suitable for your* career or not.

Also, with the basic knowledge of the C language, you can understand the flow of any program. So, now let us see what the limitations of C programming language are-

1. Concept of OOPs

C is a very vast language, but it does not support the concept of OOPs (Inheritance, Polymorphism,

Encapsulation, Abstraction, Data Hiding). C simply follows the procedural programming approach.

2. Run-time checking

In the C programming language, the errors or the bugs aren't detected after each line of code. Instead, the compiler shows all the errors after writing the program. It makes the checking of code very complex in large programs.

3. Concept of namespace C does not implement the concept of namespaces. A namespace is structured as a chain of commands to allow the reuse of names in different contexts. Without namespaces, we cannot declare two variables of the same name.

But, C programming lacks in this feature, and hence you cannot define a variable with the same name in C.

4. Lack of Exception Handling

Exception Handling is one of the most important features of programming languages. While compiling the code, various anomalies and bugs can occur. Exception Handling allows you to catch the error and take appropriate responses. However, C does not exhibit this important feature.

5. Constructor or destructor

C does not have any constructor or destructor.

Constructors & Destructors support basic functionality of Object Oriented Programming. Both are member functions that are created as soon as an object of the class is created. You will be studying constructor and destructor in detail later on.

6. Low level of abstraction

C is a small and core machine language that has minimum data hiding and exclusive visibility that affects the security of this language.

Summary

Here, we end our tutorial on ‘Advantages and Limitations of C Programming’. We hope you found this tutorial beneficial in developing a simple understanding of the pros and cons of

C. You might have understood how powerful C is in its process of implementation and execution and at the same time how it lags behind other languages in certain aspects

Modules and Functions Used

void record()
void load()
void Delay(long double)
void Move()
void Food()
void Print()
void Bend()
int Score()
void Boarder()
void Down()
void Left()
void Up()
void Right()
void ExitGame()

void record() :-

void record() function is used to record the score of the player in the file created by the game with the fopen() function. There are too many file functions used in this project like fopen(), fclose(), fprintf(), All these functions are used in file management system in C language and C++ language.

void record() Module :-

```
void record()
{
   char plname[20], nplname[20], cha, c;
   int i, j, px;
   FILE *info;
   info=fopen("record.txt","a+");
   getch();
   system("cls");
   printf("Enter your name\n");
   scanf("%[^\n]",plname);
   //************************
   for(j=0; plname[j]!='\0'; j++) //to convert
the first letter after space to capital
   {
      nplname[0]=toupper(plname[0]);
      if(plname[j-1]==' ')
      {
         nplname[j]=toupper(plname[j]);
         nplname[j-1]=plname[j-1];
      }
      else nplname[j]=plname[j];
   }
   nplname[j]='\0';
   //******************************
   //sdfprintf(info,"\t\t\tPlayers List\n");
   fprintf(info,"Player Name :%s\n",nplname);
   //for date and time

   time_t mytime;
   mytime = time(NULL);

fprintf(info,"PlayedDate:%s",ctime(&mytime))
;
   //***************************
```

```
fprintf(info,"Score:%d\n",px=Scoreonly());//ca
ll score to display score
   //fprintf(info,"\nLevel:%d\n",10);//call level
to display level
   for(i=0; i<=50; i++)
      fprintf(info,"%c",'_');
   fprintf(info,"\n");
   fclose(info);
   printf("Wanna see past records press
'y'\n");
   cha=getch();
   system("cls");
   if(cha=='y')
   {
      info=fopen("record.txt","r");
      do
      {
         putchar(c=getc(info));
      }
      while(c!=EOF);
   }
   fclose(info);
}
```

void load() :-

void load() function is used to load the initial/loading screen in the game. It gives the game’s UI a generous look for players.

void load() Module :-

```
void load()
{
   int row,col,r,c,q;
   gotoxy(36,14);
   printf("loading...");
   gotoxy(30,15);
   for(r=1; r<=20; r++)
   {
      for(q=0; q<=100000000; q++); //to
display the character slowly
      printf("%c",177);
   }
   getch();
}
```

void Delay (long double) :-

This function/module is used for delay time in this project.

void Delay(long double) Module :-

```
void Delay(long double k)
{
   Score();
   long double i;
   for(i=0; i<=(10000000); i++);
}
```

void Move() :-

This function is used to move the snakes in any direction (Up, Down, Left, Right . This is a very important module in this major project.

void Move () :-

```
void Move()
{
    int a,i;

    do
    {


        Food();
        fflush(stdin);

        len=0;

        for(i=0; i<30; i++)



{


            body[i].x=0;

            body[i].y=0;
            if(i==length)

                break;

        }

        Delay(length);

        Boarder();

        if(head.direction==RIGHT)

            Right();

        else if(head.direction==LEFT)

            Left();



else if(head.direction==DOWN)

            Down();
        else if(head.direction==UP)

           Up();

        ExitGame();
}
```

void Food() :-

void Food() function is used for food generation and consumption by the snake.

```
void Food()
{
   if(head.x==food.x&&head.y==food.y)
   {
      length++;
      time_t a;
      a=time(0);
      srand(a);
```

```
        food.x=rand()%70;
        if(food.x<=10)
            food.x+=11;
        food.y=rand()%30;
        if(food.y<=10)

            food.y+=11;
    }
    else if(food.x==0)/*to create food for the first
time coz global variable are initialized with 0*/
    {
        food.x=rand()%70;
        if(food.x<=10)
            food.x+=11;
        food.y=rand()%30;
        if(food.y<=10)
            food.y+=11;
    }
}
```

void Print() :-

void Print function is used to print the game on the console Window.

void Bend () :-

This module is used to bend the snake head in the desired direction.

int Score() :-

This function is used to store the game score in the files. These files will be stored in the system as records.

void Border() :-
This function is used to make the game border and/or wall so that the snake will not go through the wall.

void Down() :-

This function is used to move/bend the snake head in the lower/down direction.

void Up() :-

This function is used to move/bend the snake head in the upper direction.

void Left() :-

This function is used to move/bend the snake head in the Left direction.

Void Right() :-

This function is used to move/bend the snake head in the Right direction.

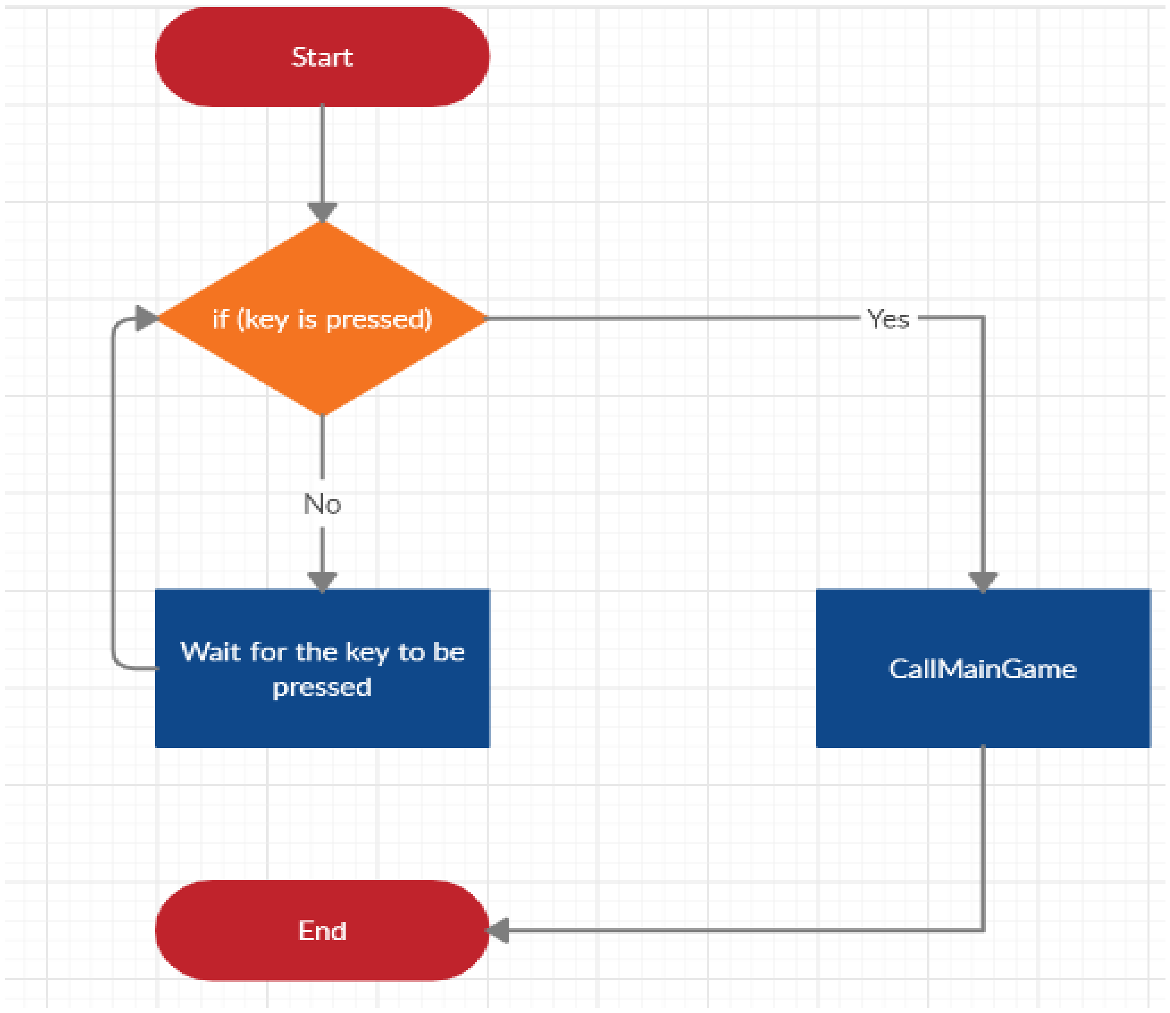
Void ExitGame() :-

This is the exit function. It is used to exit from the game.

Buttons Used

- UP
- Down
- LEFT
- RIGHT
- Press Any Button to start
- Enter Player’s Name
- Enter ‘y’ to show Previous result

Flow Chart


Start
if (key is pressed)
Yes
No
Wait for the key to be pressed
CallMainGame
End

Source Code of the Project

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <conio.h>
#include<time.h>
#include<ctype.h>
#include <time.h>
#include <windows.h>
#include <process.h>

#define UP 72
#define DOWN 80
#define LEFT 75
#define RIGHT 77

int length;
int bend_no;
int len;
```

```
char key;
void record();
void load();
int life;
void Delay(long double);
void Move();
void Food();
int Score();
void Print();
void gotoxy(int x, int y);
void GotoXY(int x,int y);
void Bend();
void Boarder();
void Down();
void Left();
void Up();
void Right();
void ExitGame();
int Scoreonly();

struct coordinate
{
   int x;
   int y;
```

```
    int direction;
};

typedef struct coordinate coordinate;

coordinate head, bend[500],food,body[30];

int main()
{

    char key;

    Print();

    system("cls");

    load();

    length=5;

    head.x=25;

    head.y=20;
```

```
   head.direction=RIGHT;

   Boarder();

   Food(); //to generate food coordinates
initially

   life=3; //number of extra lives

   bend[0]=head;

   Move();   //initialing initial bend coordinate

   return 0;

}

void Move()
{
   int a,i;

   do
   {
```

```
Food();
fflush(stdin);

len=0;

for(i=0; i<30; i++)

{

    body[i].x=0;

    body[i].y=0;

    if(i==length)

        break;

}

Delay(length);

Boarder();

if(head.direction==RIGHT)
```

```
        Right();

    else if(head.direction==LEFT)

        Left();

    else if(head.direction==DOWN)

        Down();

    else if(head.direction==UP)

        Up();

    ExitGame();

}
while(!kbhit());

a=getch();

if(a==27)
```

```
   {

       system("cls");

       exit(0);

   }
   key=getch();


if((key==RIGHT&&head.direction!=LEFT&&he
ad.direction!=RIGHT)||(key==LEFT&&head.dir
ection!=RIGHT&&head.direction!=LEFT)||(ke
y==UP&&head.direction!=DOWN&&head.dire
ction!=UP)||(key==DOWN&&head.direction!
=UP&&head.direction!=DOWN))

   {

       bend_no++;

       bend[bend_no]=head;

       head.direction=key;
```

```
    if(key==UP)
        head.y--;
    if(key==DOWN)
        head.y++;
    if(key==RIGHT)
        head.x++;
    if(key==LEFT)
        head.x--;
    Move();
}
else if(key==27)
{
        system("cls");

        exit(0);

    }

    else

    {

        printf("\a");

        Move();

    }
}

void gotoxy(int x, int y)
{

    COORD coord;

    coord.X = x;
    coord.Y = y;


SetConsoleCursorPosition(GetStdHandle(STD
_OUTPUT_HANDLE), coord);

}
void GotoXY(int x, int y)
{
    HANDLE a;
    COORD b;
    fflush(stdout);
    b.X = x;
    b.Y = y;
    a = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleCursorPosition(a,b);
}
void load()
{
    int row,col,r,c,q;
    gotoxy(36,14);
    printf("loading...");
    gotoxy(30,15);
```

```
   for(r=1; r<=20; r++)
   {
      for(q=0; q<=100000000; q++); //to
display the character slowly
      printf("%c",177);
   }
   getch();
}
void Down()
{
   int i;
   for(i=0; i<=(head.y-
bend[bend_no].y)&&len<length; i++)
   {
      GotoXY(head.x,head.y-i);
      {
         if(len==0)
            printf("v");
         else
            printf("*");
      }
      body[len].x=head.x;
      body[len].y=head.y-i;
      len++;
```

```
    }
    Bend();
    if(!kbhit())
        head.y++;
}
void Delay(long double k)
{
    Score();
    long double i;
    for(i=0; i<=(10000000); i++);
}
void ExitGame()
{
    int i,check=0;
    for(i=4; i<length; i++) //starts with 4
because it needs minimum 4 element to
touch its own body
    {

if(body[0].x==body[i].x&&body[0].y==body[i].
y)
        {
            check++;    //check's value increases as
the coordinates of head is equal to any other
body coordinate
        }
        if(i==length||check!=0)
            break;
    }

if(head.x<=10||head.x>=70||head.y<=10||h
ead.y>=30||check!=0)
    {
        life--;
        if(life>=0)
        {
            head.x=25;
            head.y=20;
            bend_no=0;
            head.direction=RIGHT;
            Move();
        }
        else
        {
            system("cls");
```

```
         printf("All lives completed\nBetter
Luck Next Time!!!\nPress any key to quit the
game\n");
         record();
         exit(0);
      }
   }
}
void Food()
{
   if(head.x==food.x&&head.y==food.y)
   {
      length++;
      time_t a;
      a=time(0);
      srand(a);
      food.x=rand()%70;
      if(food.x<=10)
         food.x+=11;
      food.y=rand()%30;
      if(food.y<=10)

         food.y+=11;
   }
```

```
    else if(food.x==0)/*to create food for the
first time coz global variable are initialized
with 0*/
    {
        food.x=rand()%70;
        if(food.x<=10)
            food.x+=11;
        food.y=rand()%30;
        if(food.y<=10)
            food.y+=11;
    }
}
void Left()
{
    int i;
    for(i=0; i<=(bend[bend_no].x-
head.x)&&len<length; i++)
    {
        GotoXY((head.x+i),head.y);
        {
            if(len==0)
                printf("<");
            else
                printf("*");
        }
        body[len].x=head.x+i;
        body[len].y=head.y;
        len++;
    }
    Bend();
    if(!kbhit())
        head.x--;

}
void Right()
{
    int i;
    for(i=0; i<=(head.x-
bend[bend_no].x)&&len<length; i++)
    {
        //GotoXY((head.x-i),head.y);
        body[len].x=head.x-i;
        body[len].y=head.y;
        GotoXY(body[len].x,body[len].y);
        {
            if(len==0)
                printf(">");
            else
            printf("*");
        }
        /*body[len].x=head.x-i;
        body[len].y=head.y;*/
        len++;
    }
    Bend();
    if(!kbhit())
        head.x++;
}
void Bend()
{
    int i,j,diff;
    for(i=bend_no; i>=0&&len<length; i--)
    {
        if(bend[i].x==bend[i-1].x)
        {
            diff=bend[i].y-bend[i-1].y;
            if(diff<0)
                for(j=1; j<=(-diff); j++)
                {
                    body[len].x=bend[i].x;
                    body[len].y=bend[i].y+j;
                    GotoXY(body[len].x,body[len].y);
                    printf("*");
                    len++;
                    if(len==length)
                        break;
                }
            else if(diff>0)
                for(j=1; j<=diff; j++)
                {
                    /*GotoXY(bend[i].x,(bend[i].y-j));
                    printf("*");*/
                    body[len].x=bend[i].x;
                    body[len].y=bend[i].y-j;
                    GotoXY(body[len].x,body[len].y);
                    printf("*");
                    len++;
                    if(len==length)
                        break;
                }
        }
        else if(bend[i].y==bend[i-1].y)
        {
            diff=bend[i].x-bend[i-1].x;
            if(diff<0)
                for(j=1; j<=(-diff)&&len<length; j++)
                {
                    /*GotoXY((bend[i].x+j),bend[i].y);
                    printf("*");*/
                    body[len].x=bend[i].x+j;
                    body[len].y=bend[i].y;
                    GotoXY(body[len].x,body[len].y);
                    printf("*");
                    len++;
                    if(len==length)
                        break;
                }
            else if(diff>0)
                for(j=1; j<=diff&&len<length; j++)
                {
                    /*GotoXY((bend[i].x-j),bend[i].y);
                    printf("*");*/
                    body[len].x=bend[i].x-j;
                    body[len].y=bend[i].y;
                    GotoXY(body[len].x,body[len].y);
                    printf("*");
                    len++;
                    if(len==length)
                        break;
                }
```

```
        }
    }
}
void Boarder()
{
    system("cls");
    int i;
    GotoXY(food.x,food.y);   /*displaying
food*/
    printf("F");
    for(i=10; i<71; i++)
    {
        GotoXY(i,10);
        printf("!");
        GotoXY(i,30);
        printf("!");
    }
    for(i=10; i<31; i++)
    {
        GotoXY(10,i);
        printf("!");
        GotoXY(70,i);
        printf("!");
    }
```

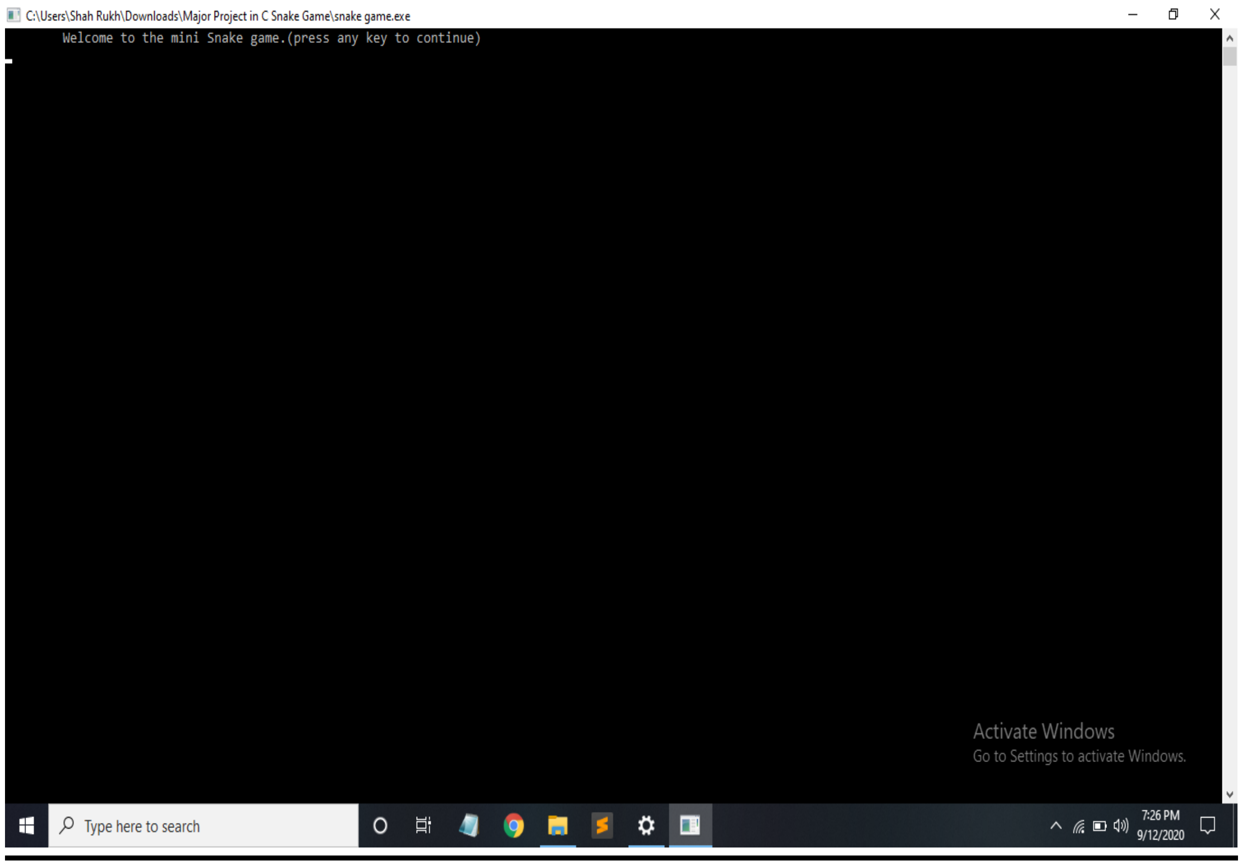
```
}
void Print()
{
   //GotoXY(10,12);
   printf("\tWelcome to the mini Snake
game.(press any key to continue)\n");
   getch();
   system("cls");
   printf("\tGame instructions:\n");
   printf("\n-> Use arrow keys to move the
snake.\n\n-> You will be provided foods at
the several coordinates of the screen which
you have to eat. Everytime you eat a food the
length of the snake will be increased by 1
element and thus the score.\n\n-> Here you
are provided with three lives. Your life will
decrease as you hit the wall or snake's
body.\n\n-> YOu can pause the game in its
middle by pressing any key. To continue the
paused game press any other key once
again\n\n-> If you want to exit press esc. \n");
   printf("\n\nPress any key to play game...");
   if(getch()==27)
      exit(0);
```

```
}
void record()
{
   char plname[20],nplname[20],cha,c;
   int i,j,px;
   FILE *info;
   info=fopen("record.txt","a+");
   getch();
   system("cls");
   printf("Enter your name\n");
   scanf("%[^\n]",plname);
   //************************
   for(j=0; plname[j]!='\0'; j++) //to convert
the first letter after space to capital
   {
      nplname[0]=toupper(plname[0]);
      if(plname[j-1]==' ')
      {
         nplname[j]=toupper(plname[j]);
         nplname[j-1]=plname[j-1];
      }
      else nplname[j]=plname[j];
   }
   nplname[j]='\0';
   //*****************************
   //sdfprintf(info,"\t\t\tPlayers List\n");
   fprintf(info,"Player Name :%s\n",nplname);
   //for date and time

   time_t mytime;
   mytime = time(NULL);
   fprintf(info,"Played
Date:%s",ctime(&mytime));
   //****************************

fprintf(info,"Score:%d\n",px=Scoreonly());//ca
ll score to display score
   //fprintf(info,"\nLevel:%d\n",10);//call level
to display level
   for(i=0; i<=50; i++)
      fprintf(info,"%c",'_');
   fprintf(info,"\n");
   fclose(info);
   printf("Wanna see past records press
'y'\n");
   cha=getch();
   system("cls");
   if(cha=='y')
```

```
    {
        info=fopen("record.txt","r");
        do
        {
            putchar(c=getc(info));
        }
        while(c!=EOF);
    }
    fclose(info);
}
int Score()
{
    int score;
    GotoXY(20,8);
    score=length-5;
    printf("SCORE : %d",(length-5));
    score=length-5;
    GotoXY(50,8);
    printf("Life : %d",life);
    return score;
}
int Scoreonly()
{
    int score=Score();
    system("cls");
    return score;
}
void Up()
{
    int i;
    for(i=0; i<=(bend[bend_no].y-
head.y)&&len<length; i++)
    {
        GotoXY(head.x,head.y+i);
        {
            if(len==0)
                printf("^");
            else
                printf("*");
        }
        body[len].x=head.x;
        body[len].y=head.y+i;
        len++;
    }
    Bend();
    if(!kbhit())
        head.y--;
}
```

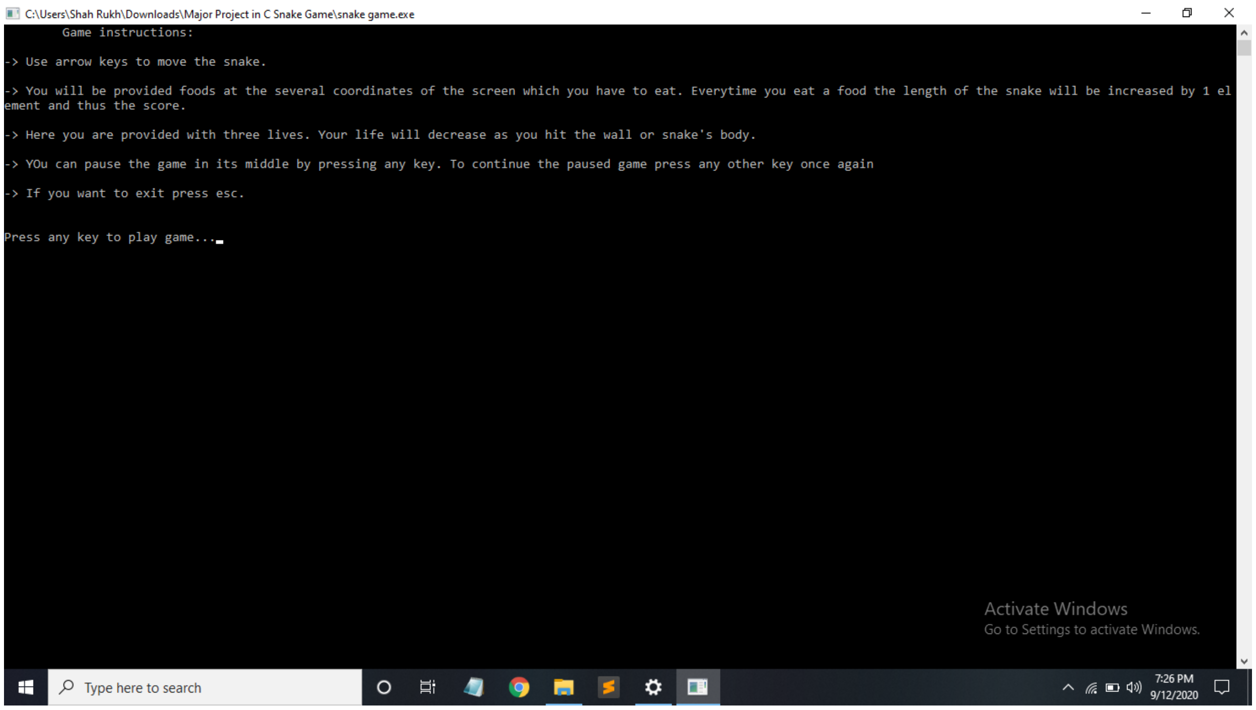
Expected Result

Initial Loading screen
Press any key to start the game.


C:\Users\Shah Rukh\Downloads\Major Project in C Snake Game\snake game.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:26 PM
9/12/2020

```
      Welcome to the mini Snake game.(press any key to continue)
```

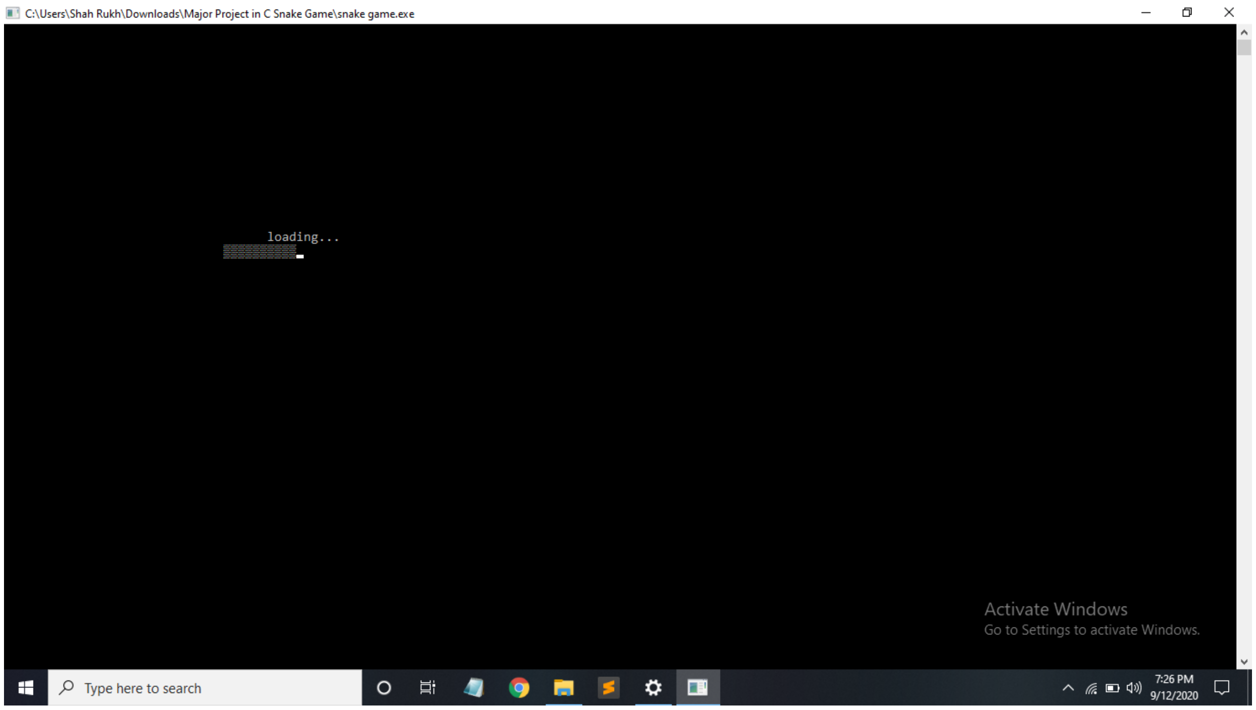
Game instruction screen



C:\Users\Shah Rukh\Downloads\Major Project in C Snake Game\snake game.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:26 PM
9/12/2020

```
        Game instructions:

-> Use arrow keys to move the snake.

-> You will be provided foods at the several coordinates of the screen which you have to eat. Everytime you eat a food the length of the snake will be increased by 1 el
ement and thus the score.

-> Here you are provided with three lives. Your life will decrease as you hit the wall or snake's body.

-> YOu can pause the game in its middle by pressing any key. To continue the paused game press any other key once again

-> If you want to exit press esc.


Press any key to play game...
```

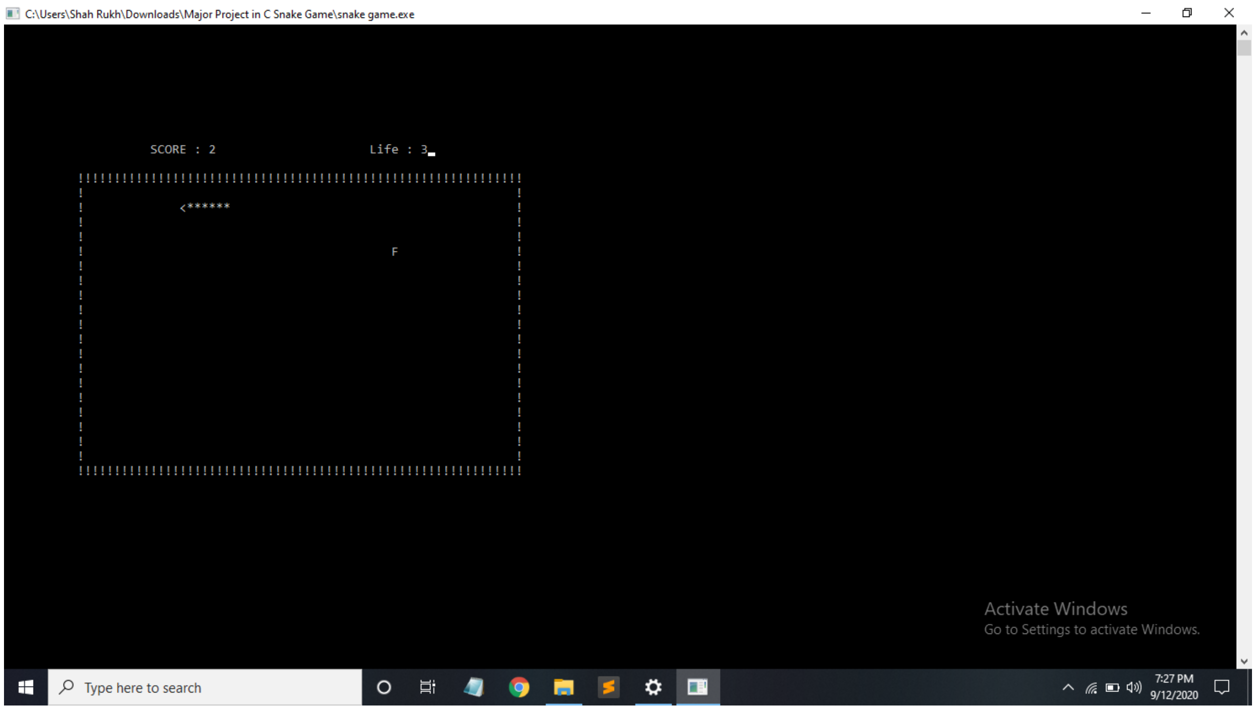
Loading Screen



C:\Users\Shah Rukh\Downloads\Major Project in C Snake Game\snake game.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:26 PM
9/12/2020

```
      loading...
```

Actual Game Screen

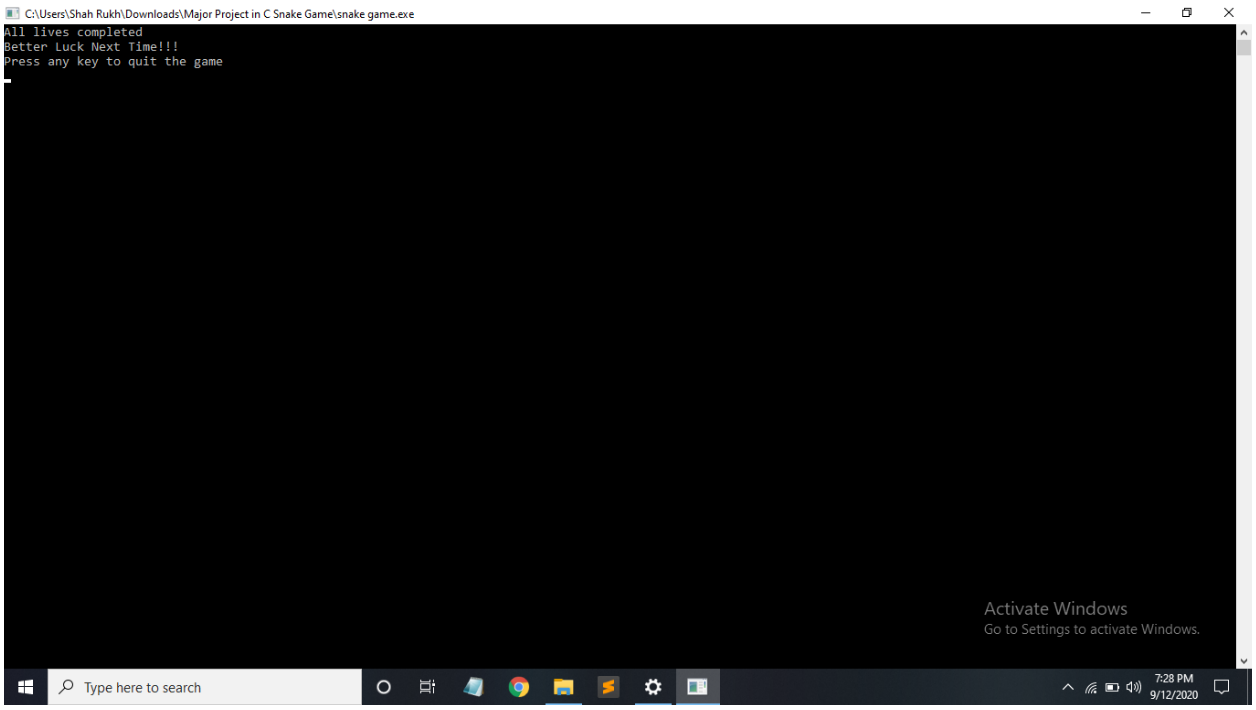


```
C:\Users\Shah Rukh\Downloads\Major Project in C Snake Game\snake game.exe

SCORE : 2                    Life : 3
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                                           !
!            <******                                        !
!                                                           !
!                                                           !
!                                       F                   !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Activate Windows
Go to Settings to activate Windows.

Type here to search
7:27 PM
9/12/2020

Game Over Screen, Press any key to quit the game.

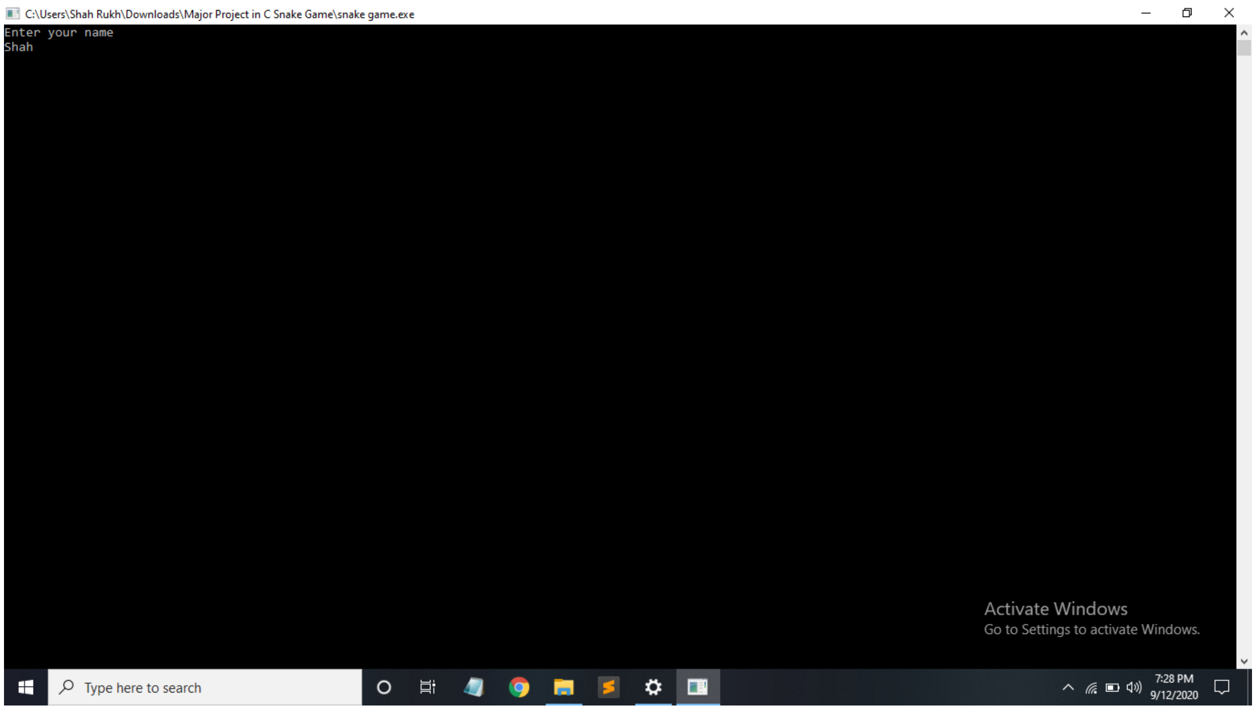


```
C:\Users\Shah Rukh\Downloads\Major Project in C Snake Game\snake game.exe
All lives completed
Better Luck Next Time!!!
Press any key to quit the game
```

Activate Windows
Go to Settings to activate Windows.

Type here to search
7:28 PM
9/12/2020

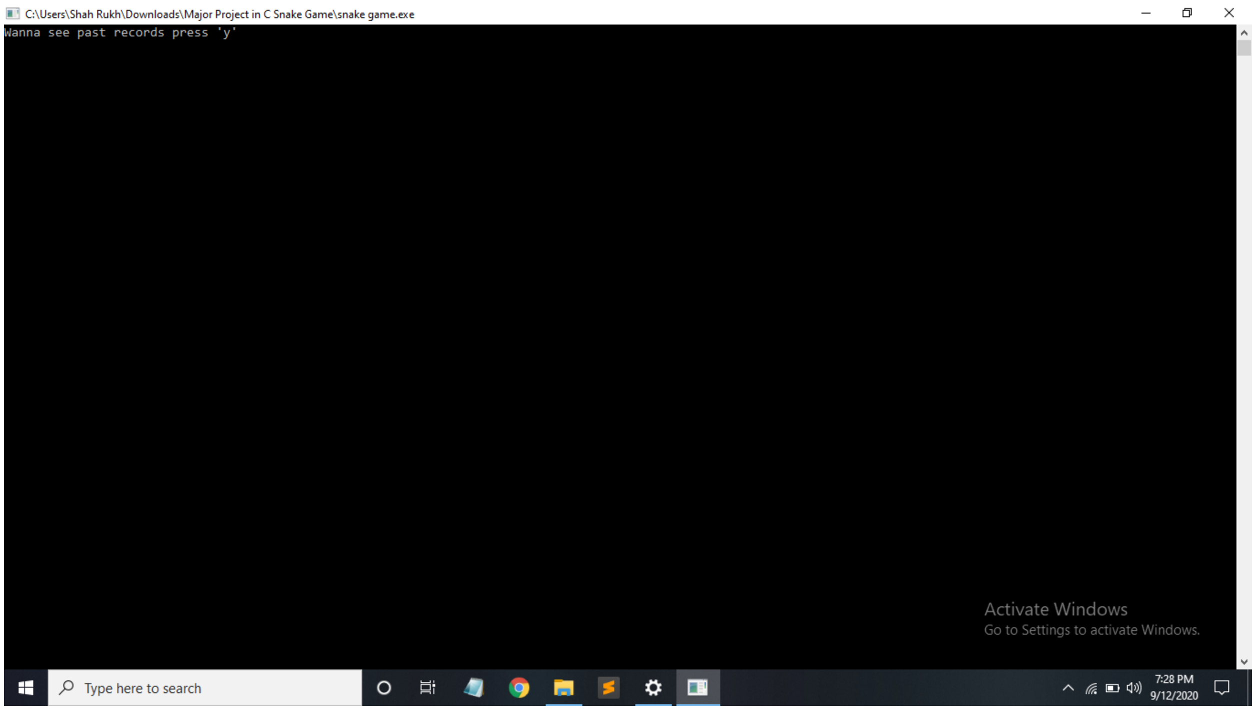
Enter player name if you want.


C:\Users\Shah Rukh\Downloads\Major Project in C Snake Game\snake game.exe

```
Enter your name
Shah
```

Activate Windows
Go to Settings to activate Windows.
Type here to search
7:28 PM
9/12/2020

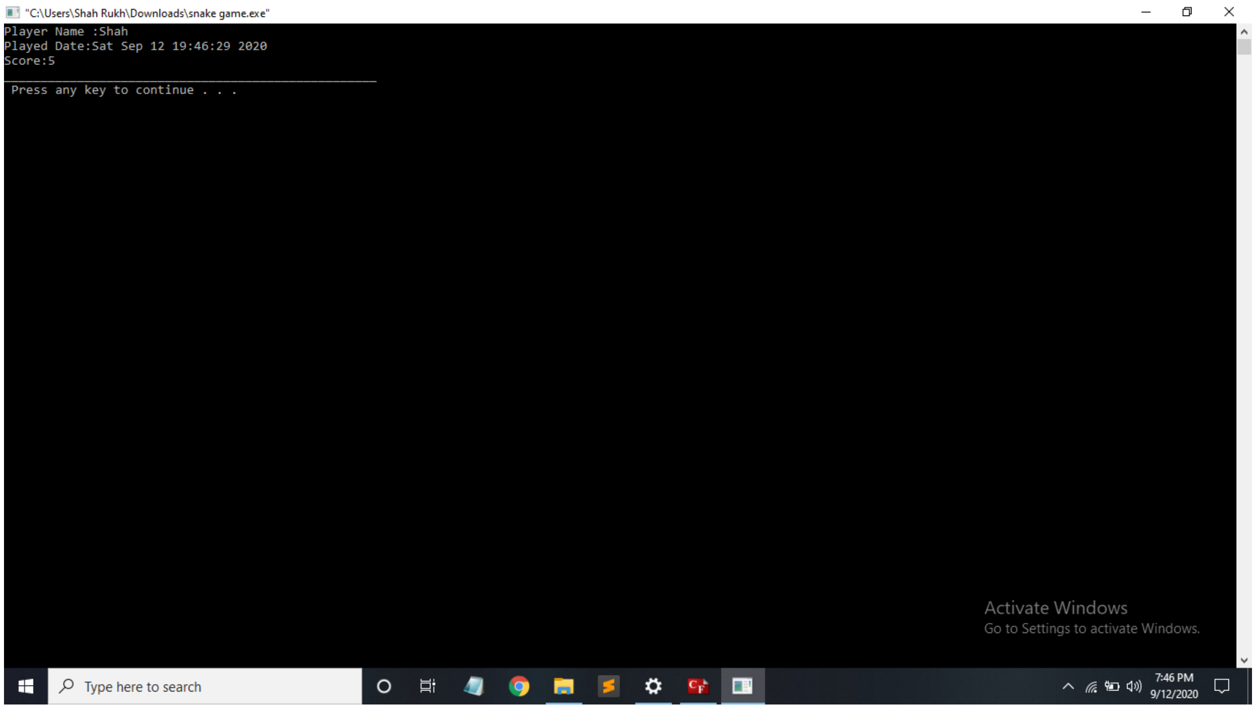
Press ‘y’ to see the score.


C:\Users\Shah Rukh\Downloads\Major Project in C Snake Game\snake game.exe

```
Wanna see past records press 'y'
```

Activate Windows
Go to Settings to activate Windows.
Type here to search
7:28 PM
9/12/2020

Previous score screen.


"C:\Users\Shah Rukh\Downloads\snake game.exe"
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:46 PM
9/12/2020

```
Player Name :Shah
Played Date:Sat Sep 12 19:46:29 2020
Score:5
_____________________________________________
 Press any key to continue . . .
```

The final version of the proposed game will deliver the following features:

1. A game that will retain the simplicity of snake traditional snake game and contain attractive graphics and user interface to attract the players.
2. The real time experience of commercial multiplayer games will be available in the snake game that will allow more than one players to play a game simultaneously over a network.
3. A computer controlled opponent – Snake that will strongly challenge the human players in the game. This functionality will make the game more interesting and challenging. This will also be a unique feature of the game over existing versions of snake game.

References

- Let us C 15th Edition by Yashavant Kanetkar
- Programming in ANSI C 6th Edition by E. Balagurusamy
- C in Depth – S.K Srivastava
- https://www.codewithc.com
- https://data-flair.training
- https://www.atnyla.com
- https://www.tutorialspoint.com
- https://app.creately.com