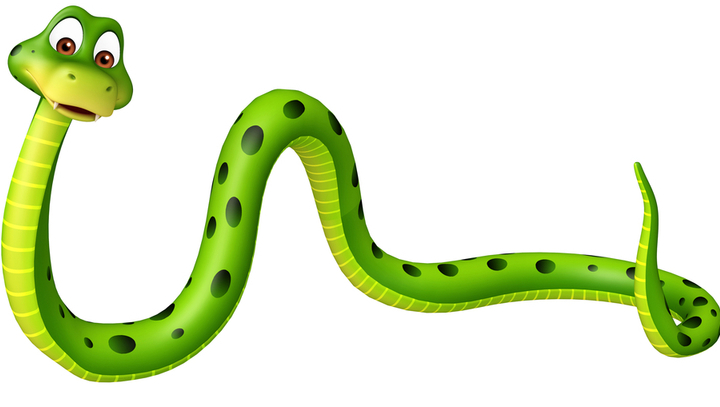
**CONTENTS**

* **Introduction of Project**
* **Reqirements**
* **Source Code**
* **Output and Result Of the project**
* **Conclusion**
* **References**

**INTRODUCTION**



*Fig1.snake*

The Snake Game In C Programming With Source Code is developed using C programming language. This snake game in c is a basic console program with no graphics. And this snake game in c language is good for the students or beginners who wants to learn programming especially in C Programming Language

This Mini Project in C Snake Game is a simple console application without graphics. In this project, you can play the popular “Snake Game” just like you played it elsewhere. You have to use the up, down, right or left arrows to move the snake.Foods are provided at the several co-ordinates of the screen for the snake to eat. Every time the snake eats the food, its length will by increased by one element along with the score.

The history of the Snake game goes back to the 1970's. However, it was the 1980's when the game took on the look that we will be using. It was sold under numerous names and many platforms but probably gained widespread recognition when it was shipped as standard on Nokia mobile phones in the late 1990's.

The game involves controlling a single block or snakehead by turning only left or right by ninety degrees until you manage to eat an apple. When you get the apple, the Snake grows an extra block or body segment.

If, or rather when, the snake bumps into the edge of the screen or accidentally eats himself the game is over. The more apples the snake eats the higher the score.

The player controls a long, thin creature, resembling a snake, which roams around on a bordered plane, picking up food (or some other item), trying to avoid hitting its own tail or the edges of the playing area. Each time the snake eats a piece of food, its tail grows longer, making the game increasingly difficult.

A snake game in c language code is an older classic video game. The Game was first created in late 70s. Later it was brought to PCs. In this snake game program in c language, the player controls a snake. The objective of this snake game using c is to eat as many as possible. The snake must avoid the walls and its own body

**REQIREMENTS**

**Hardware reqirements**

* Hp laptop i5 core with AMD RADEON graphics card.
* Ram 8 GB Rom 512 GB

**Operating System**

* Windows 11

**Software reqirements**

* For this project code we are using ***DEV C++*** software (IDE)
* Dev-C++ allows you to write, compile and run a C or C++ program. C++ programming language is an enhanced version of C language that provides object-oriented programming (OOP) capabilities. It is a superset of C, which means that you can use a C++ compiler to compile C programs.

**Header files**

* #include <stdio.h> function is a header file which has the necessary information to include the input/output related functions in our program.
* #include <conio.h> It is used to include the console input output library functions.
* #include <stdlib.h> is the header of the general purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions and others.
* #include <process.h>is a C header file which contains function declarations and macros used in working with threads and processes
* #include <windows.h> header file is used to access the Win32 API functions and it makes it easier for the user to use the in-built functionality

**Functions**

* void record();
* void load( );
* 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( );
* In this program we are also use the various Keywords and loops for example :
* If else loop , while loop , do while loop,etc.

**SOURCE CODE**

**FEW CODE LINES OF PROJECT SNAKE GAME IN C :**

#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);

P)

head.y--;

}

}

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);

}

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 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--;

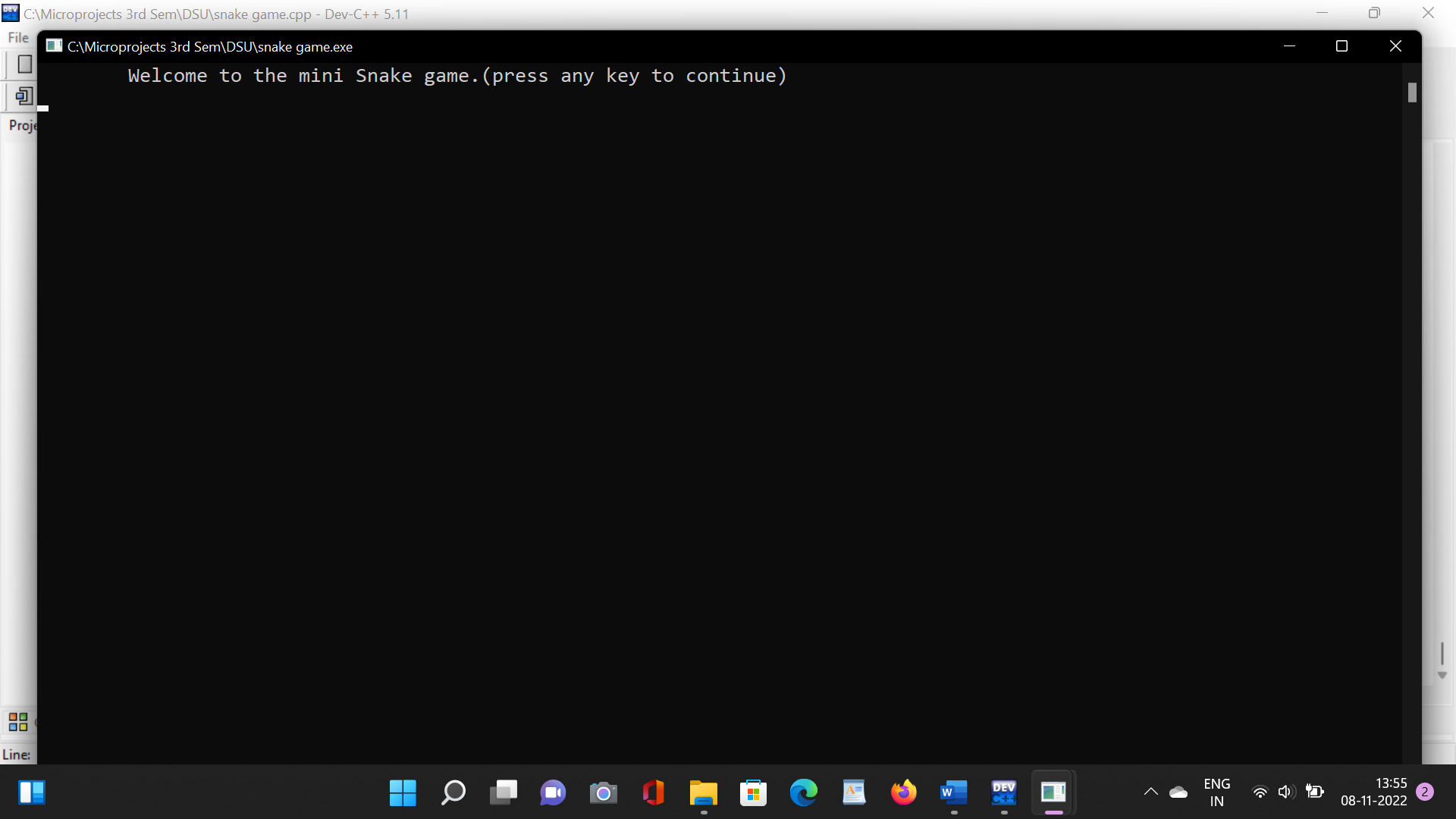
}

**FULL SOURCE CODE here :**

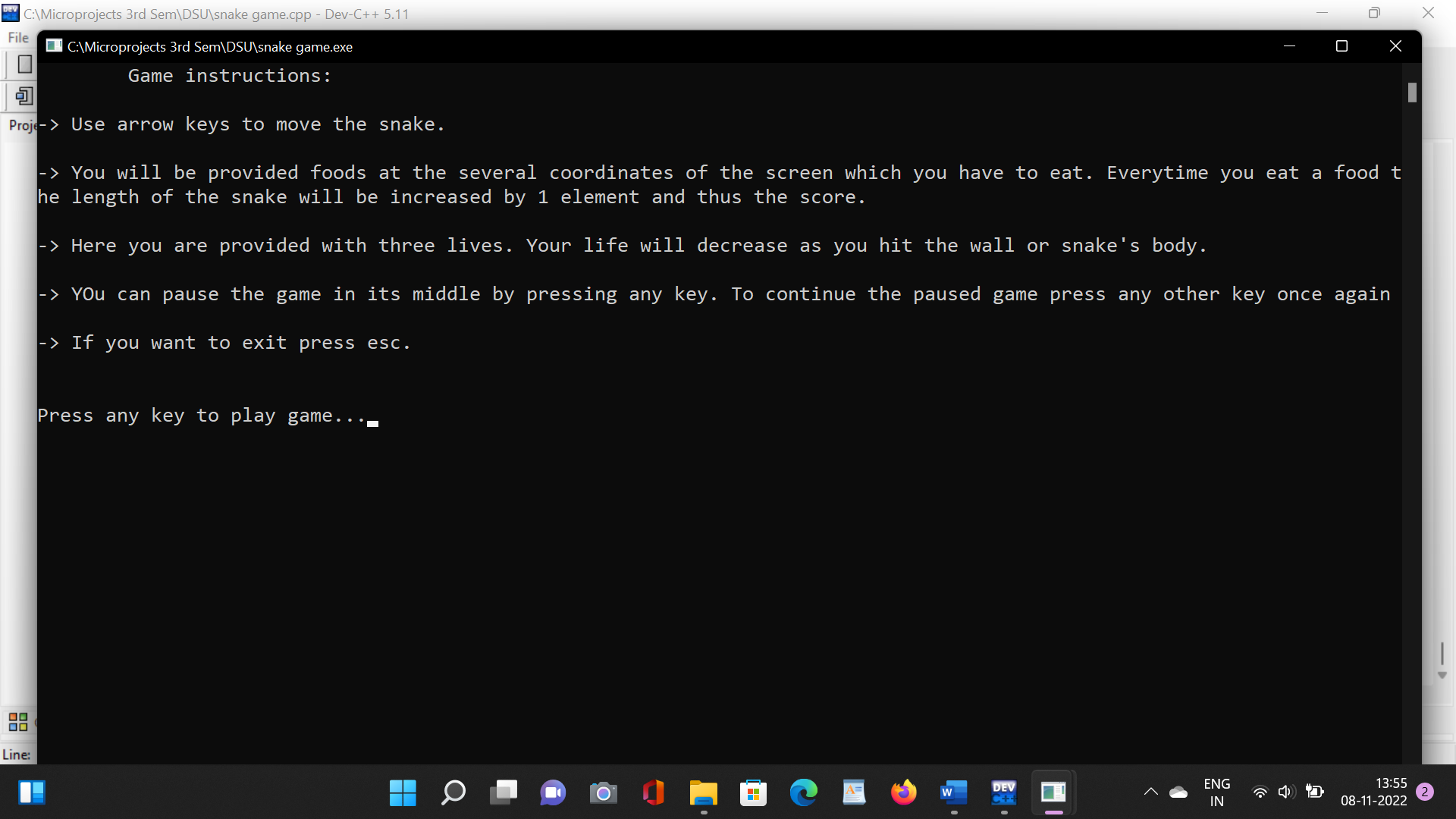


**RESULT AND OUTPUT**

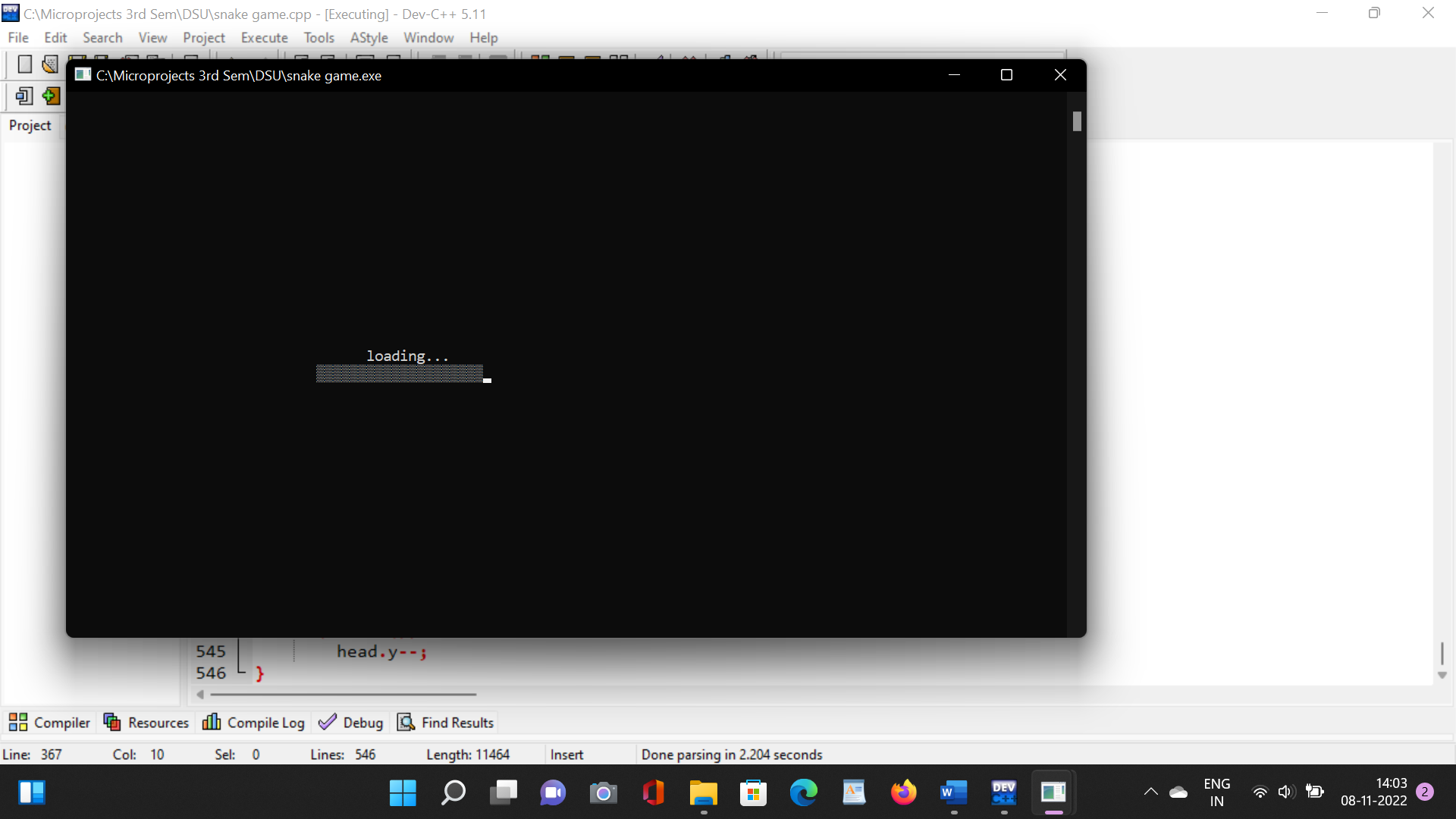
**OUTPUT ON SCREEN**

****

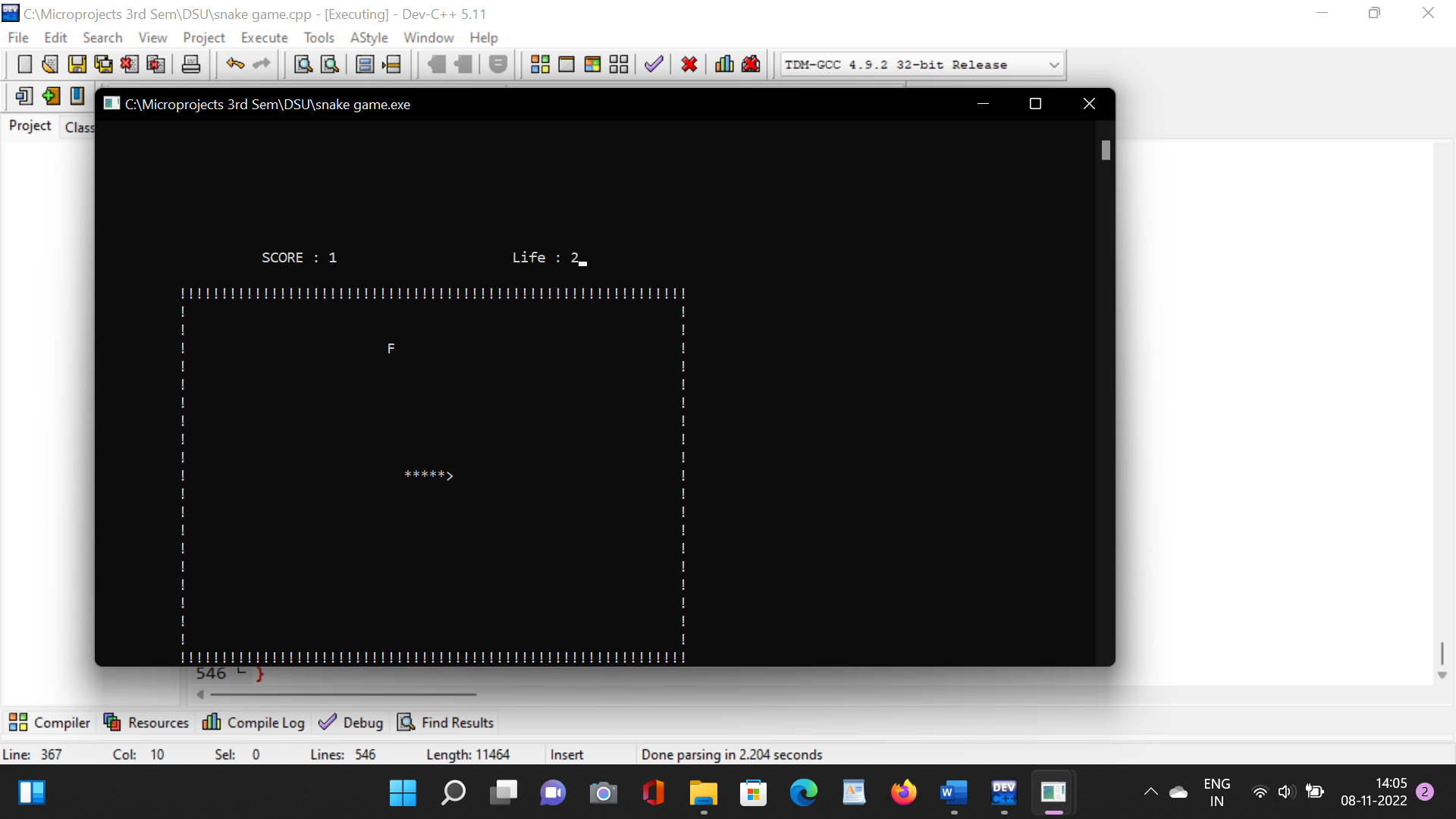
***Fig2. Screen 1***



**Fig3.screen 2 ( GAME INSTRUCTIONS )**

****

**Fig4.screen 3 ( LOADING GAME )**

****

**Fig5.screen 4 ( GAME START ) play & enjoy**

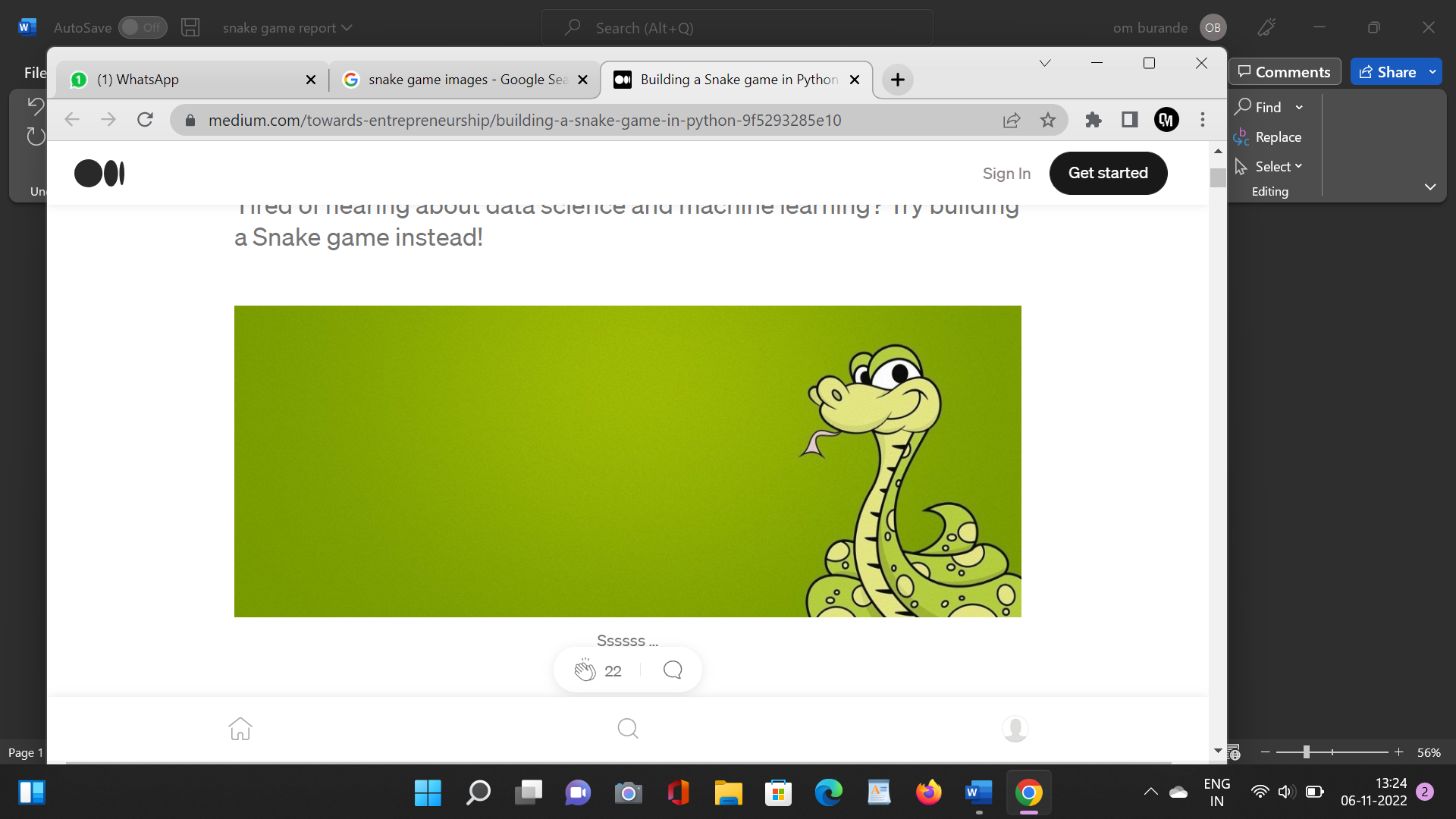
**CONCLUSION**

Snake game is a computer action game , whose goal is to control a snake to move and collect food in a map. In this paper we develop a controller based on movement rating functions considering smoothness , space , and food.

While the final version resulted in a snake game that could eat food , the movement glitch caused the food to cause further size issues.Despite the fact that the game could not truly be played due to the fact no score could be given , the game is still satisfying.

The objective of this c project is to build a snake game project. In this c project ,the player has to move a snake touches the fruit. If the snake touches itself three times or the border of the game then the game will over and you are loose the game. If you want to play again then press enter two times to start the game.

This is very intresting game and development of the game is also easy not difficult In this project or report we clear all things (points) about the snake game project.



**REFERENCES**

[**https://www.geeksforgeeks.com**](https://www.geeksforgeeks.com)

[**https://www.studytonight.com**](https://www.studytonight.com)

[**https://www.slideshare.net**](https://www.slideshare.net)

[**https://www.Javatpoint.com**](https://www.Javatpoint.com)

[**https://myprojectideas.com**](https://myprojectideas.com)

[**https://quora.com**](https://quora.com)

[**https://w3school.com**](https://w3school.com)

**THANK YOU…**