**Noida Institute of Engineering and Technology**

Department of Computer Science and Engineering



**MINI – PROJECT (KCS-354)**

**Classic Snake Game**

Submitted by: Submitted to:

Karan Kanojia (1901330100132) Mrs. Ankita

Jeetesh kumar (1901330100128) Sharma

Jeevansh Khanna (1901330100129) (department of

Khushee Gupta (1901330100135) computer science)

[Group -G5]

CS-3-B

**CERTIFICATE**

This is to certify that the mini project report entitled “Classic Snake Game” is a record of the work done by the following student:

Karan Kanojia (1901330100132)

Jeetesh kumar (1901330100128)

Jeevansh Khanna (1901330100129)

Khushee Gupta (1901330100135)

This work is done under our supervision and guidance during the academic year of 2019-2020. This report is submitted to the NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY (NIET), GREATER NOIDA for partial fulfillment for a degree in Bachelors of Technology in Computer Science and Engineering from Dr. APJ Abdul Kalam Technical University, Uttar Pradesh, India.

We wish all their future endeavors’.

Signature of the mentor

Mrs. Ankita Sharma

**Acknowledgement**

I would like to place on record my deep sense of gratitude to Mrs. Ankita Sharma in the Department of Computer Science and Engineering, Noida Institute of Engineering & Technology, Greater Noida, Gautam Budha Nagar, Uttar Pradesh, India for his/her generous guidance, help and useful suggestions.

I express my sincere gratitude to Prof. Chandra Shekhar Yadav, HOD(CSE), Noida Institute of Engineering & Technology, Greater Noida for his stimulating guidance, Continuous encouragement and supervision throughout the course of present work.

**INDEX**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S. no | Title | Page number |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 1 | Cover page |  |  |  |  |  |  |  |  |
| 2 | Certificate |  |  |  |  |  |  |  |  |
| 3 | Acknowledgement |  |  |  |  |  |  |  |  |
| 4 | Introduction |  |  |  |  |  |  |  |  |
| 5 | Objective |  |  |  |  |  |  |  |  |
| 6 | Literature review |  |  |  |  |  |  |  |  |
| 7 | History |  |  |  |  |  |  |  |  |
| 8 | Sample image |  |  |  |  |  |  |  |  |
| 9 | Methodology |  |  |  |  |  |  |  |  |
| 10 | Plan of work |  |  |  |  |  |  |  |  |
| 11 | Modules |  |  |  |  |  |  |  |  |
| 12 | Tools required |  |  |  |  |  |  |  |  |
| 13 | Screenshot |  |  |  |  |  |  |  |  |
| 14 | Coding phase |  |  |  |  |  |  |  |  |
| 15 | Conclusion |  |  |  |  |  |  |  |  |
| 16 | Future scope |  |  |  |  |  |  |  |  |
| 17 | References |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**Synopsis**

**INTRODUCTION**

This SNAKE GAMING C project deals with the snake game. In this snake going to eat objects randomly emerging on screen and if successful in eating then it becomes larger in size and gains score. The player has to change the direction of the snake by pressing left, right, top, down arrows for getting the food .Addition feature is completion of the game in a given time which makes game addictive and has ability to mesmerize the player. This is small & simple C Programming application for game lovers & programming fresher’s. This project was developed in Turbo C Programming Language & No additional databaserequired*.*

**OBJECTIVE**

This game aims to change the way people think of traditional snake game.

It will offer the Experience of commeral games to the player

 retaining the simplicity of traditional Snake game.

The major objectives of this project are:

* Create a snake game that will have all the functionality of

traditional snake games.

* Introduce computer controlled intelligent opponent (unique feature of this game) to Make the game more challenging and interesting. The movement and action of these Intelligent opponents will be controlled by computer whose aim will be to eat the food Before human players capture it.

**LITERATURE REVIEW**

Snake is a video game concept which originated during the late 1970s in arcades. The name applies to the general game design; the original was not named Snake, and there is no definitive version of the game. Its simplicity has led to many implementations of the Snake concept. After it became the standard pre-loaded game on Nokia mobile phones in 1998, there was a resurgence of interest in the game as it found a larger audience.

**HISTORY**

The first published by Nokia, for monochrome phones. It was programmed in 1997 by Taneli Armanto of Nokia and introduced on the Nokia 6110.

Sone other versions of snake games are :-

* Snake II \* Snake EX2
* Snake Xenzia \* Snakes Subsonic
* Snake EX
* Snake III

**IMAGE SHOWING SNAKE GAME IN NOKIA PHONE**

**

**METHODOLOGY**

* **SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)**

The stages of SDLC are as follows:

**Stage 1: Planning and requirement analysis**

Requirement Analysis is the most important and necessary stage in SDLC.

The members of the team perform it with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.

**Stage 2: Defining Requirements**

This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

**Stage 3: Designing the Software**

The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project.

**Stage 4: Developing the project**

In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concern.

**Stage 5: Testing the Product**

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested.

**Stage 6: Deployment**

Once the product is tested and ready , then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.

After the software is deployed, then its maintenance begins.

**Stage7: Maintenance**

Once when the user starts using the developed systems, then the real issues come up and requirements to be solved from time to time.

This procedure where the care is taken for the developed product is known as maintenance.



**PLAN OF WORK**

* **Pert chart**

**Synopsis**

**Coding**

**Merge task**

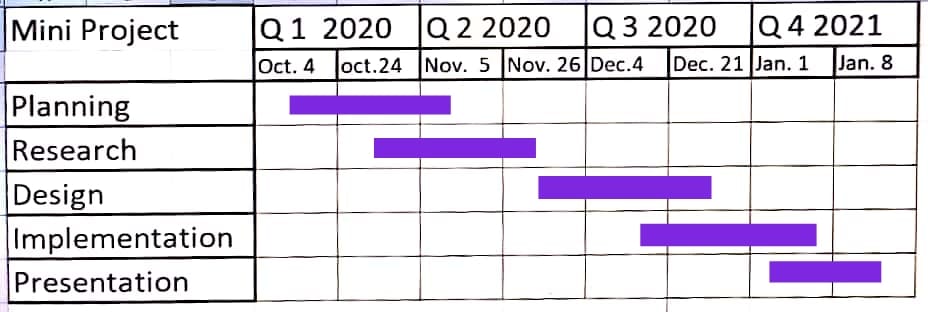
**Info. *Gathering***

**Final report**

**Topic selection**

**Coding**

**Design**

* **Gantt chart**

**MODULES**

Basically this classic snake game have only one module (user module) but is sub divided into several sub-modules, as discussed below:-

* **USER MODULE :-**
* **Control –** player have to control the moving snake by pressing **left, right, up,** & **down** keys.
* **Food –** User have to eat up the food as much as he can.
* **Length –** As the snake will eat the food, randomly emerging on the screen, the length of the snake will be automatically increase by 1.
* **Score –** If the snake touches it’s body or the boundaries surrounding on it**.** Then the user will out and the game will be ended and the final score will get displayed on the user’s screen.

**TOOLS & TECHNOLOGIES REQUIRED**

**HARDWARE**

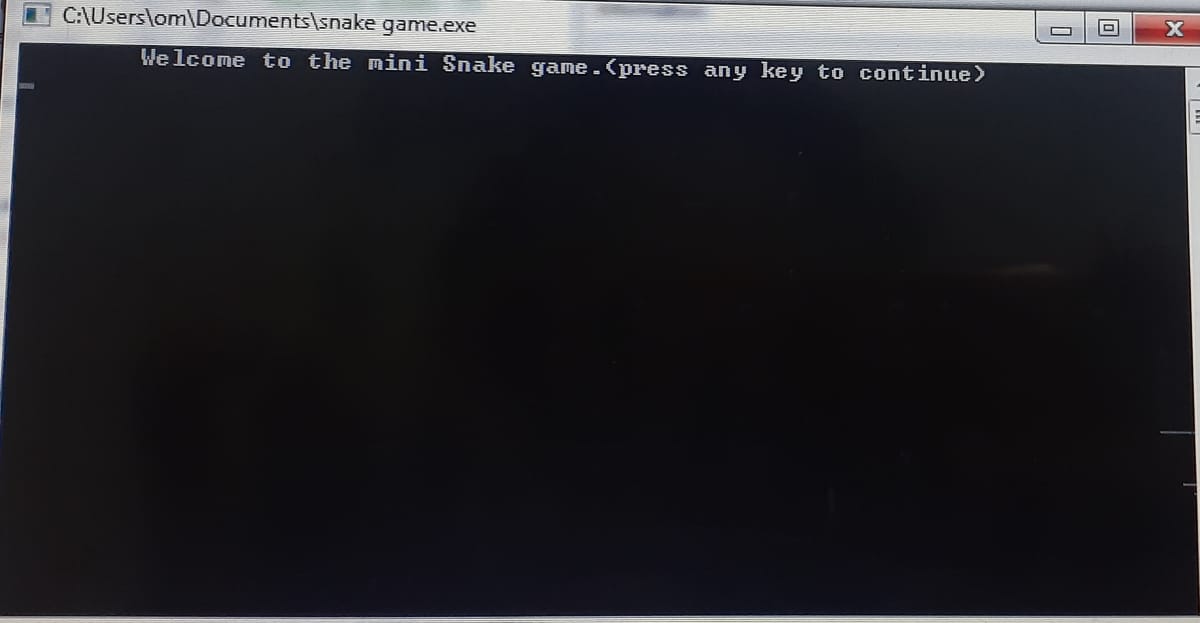
* Hard disk – 32 GB
* RAM – 128 MB, 512 MB

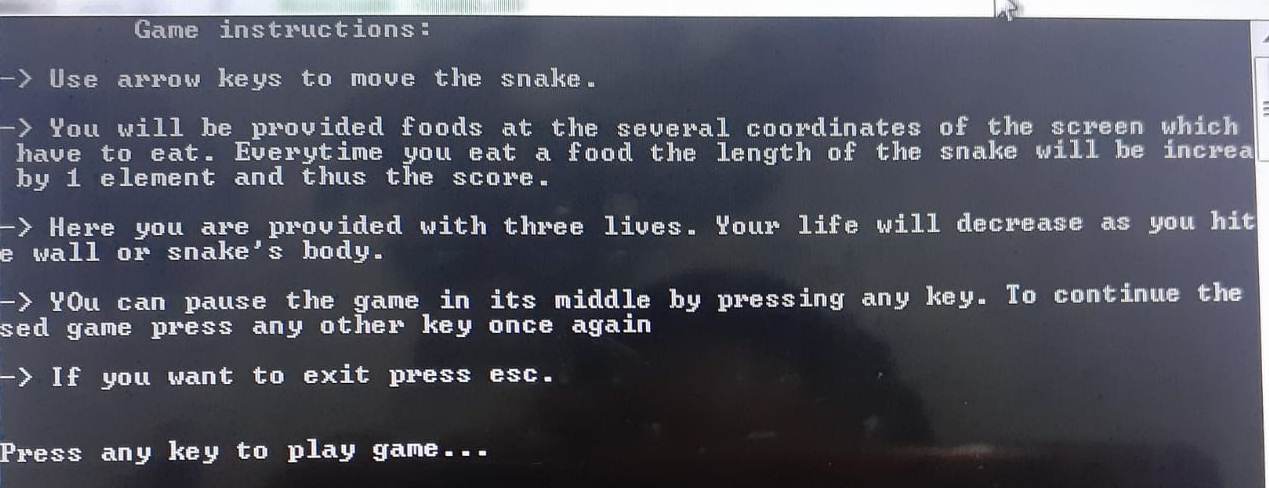
**SOFTWARE**

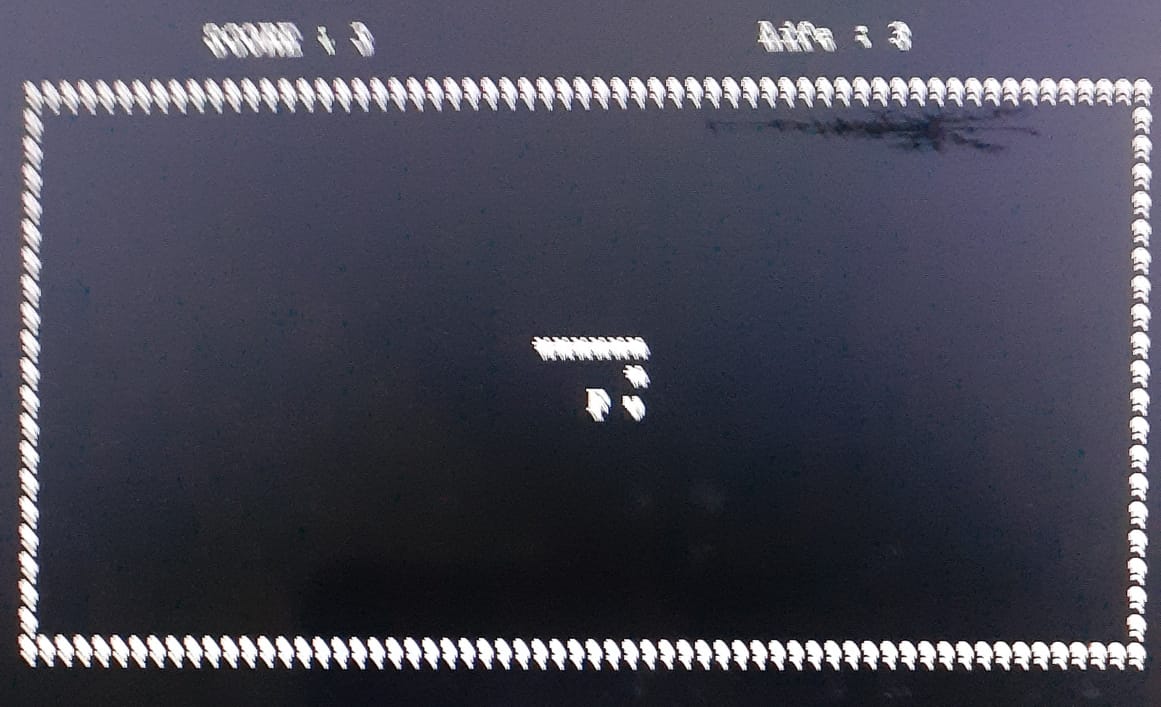
The main thing you must remember that you must have any C complier (Borland turbo C,GCC, Dev c/c++) in your Laptop, Computer and in you mobile phone.

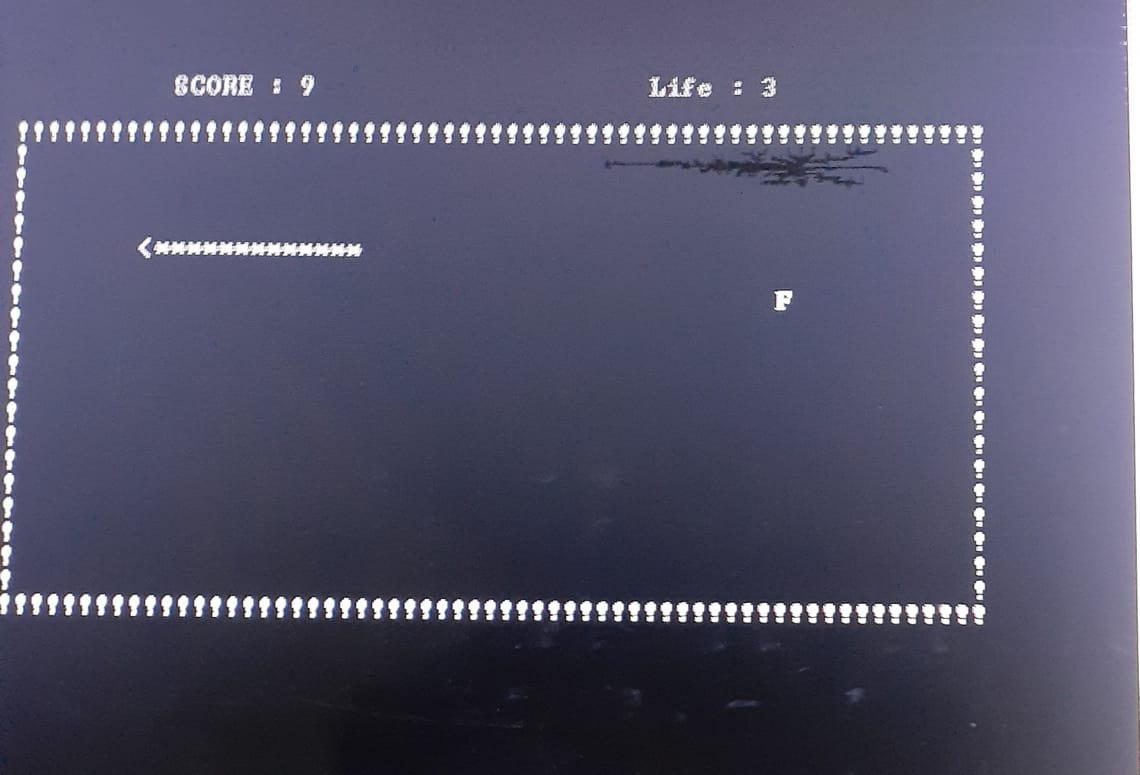
* Operating system – Windows XP / Window Vista
* Application software – Turbo C/C++ 2.0
* Language – C
* Processor — Pentium IV processor

**SCREENSHOT OF SNAKE GAME**

****

****

****

****

**CODING PHASE**

#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&&head.direction!=RIGHT)||(key==LEFT&&head.direction!=RIGHT&&head.direction!=LEFT)||(key==UP&&head.direction!=DOWN&&head.direction!=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||head.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());//call 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--;

}

**CONCLUSION**

This project gives us more thrill, adventure and also gives us more pleasure. It helps us in many sectors like- planning, designing, developing, managing, programming skill, socket programming and so on*.*

**FUTURE SCOPE**

Our project will be able to implement I’m future after making some changes and medication as we make our project at a basic or average level.

So the modifications that can be done in our project are:

* It can be made with good graphics.
* We can add more options like Top score, Player profile,, etc.
* We can add multi-players options.
* We can also add background sound.
* It can be used to demonstrate the students, as an example for C programming.

**REFERENCE**

.

<http://www.c-sharpcorner.com/UploadFile/udeshikah/snake-game-application-in-C-Sharp/>

<http://www.dreamincode.net/forums/topic/243537-control-issue-with-snake-game-in-c%23/>

http://csharp.net-informations.com/communications/csharp-chat-client.htm

**BOOK’S REFERENCES**

1. The Java Programming Language, by Ken Arnold, James Gosling, David Holmes
2. Java Threads, Second Edition, by Scott Oaks, Henry Wong
3. Core Java-Volume I/II, Seventh Edition, by Cay S. Horstmann, Gary Cornell

***Thankyou***