**INTRODUCTION:**

The Snake game project is a console-based simple project which is designed without using graphics.

While playing this game you will feel the realistic feel of mobile snake game. Which is as same as the mobile snake game only the difference is that the graphics are used in mobile game and we have not used graphics in this project.

**DESCRIPTION:**

Snake is a video game genre where the player maneuvers a growing line that becomes a primary obstacle to itself.

The food for the snake is provided at various co-ordinates of the screen to eat. While playing the game, when the snake eats the food automatically the length of the snake is increased by one element. And the score is also increased when the snake eats the food.

In this game, only 3 lives are given to score the points after three life the game will terminate automatically.

The food is represented with a F(alphabet) symbol.

The snake is represented with an \*(asterisk) symbol.

The snake can move in any direction according to the user with the help of the keyboard (Arrow keys).

When the snake eats a fruit the score will increase by 1 point.

The fruit will generate automatically within the boundaries.

Whenever the snake will touch the boundary the game is over.

**Built-in functions used**:

**Kbhit():** This function in C Is used to determine if a key has been pressed or not. To use this function in a program include the header file

**Conio.h**

. If a key has been pressed, then it returns a non-zero value otherwise it returns zero.

**Stdlib.h**

. It returns a random integer value every time it is called.

**CODE:**

//LEARNPROGRAMO-PROGRAMMING MADE SIMPLE

#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:**

Snake game is a computer action game, whose goal is to control a snake to move and collect food in a map. It is played for the entertainment purpose.

**OUTPUT:**