1. Introduction

The project “ Snake Game”isgoing to be developed for PU school of science and technology PUSAT , Biratnagar, provence -01, Nepal.Snake gameis a computer action game and the player’s 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. Scores given by these functions are aggregated by linear weighted sum, and the snake takes the action that leads to the highest score. To find a set of good weight values, we apply an evolutionary algorithm. We examine several algorithm variants of different crossover and environmental selection operators.

**2. Objective**

Snake game is one of the most popular arcade games of all time. In this game, the main objective of the player is to catch the maximum number of fruits without hitting the wall or itself. Creating a snake game can be taken as a challenge while learning C or Cgame. It is one of the best beginner-friendly projects that every novice programmer should take as a challenge. Learning to build a game is kind of interesting and fun learning.

3. Features

The player controls a dot, square, or object on a bordered plane. As it moves forward, it leaves a trail behind, resembling a moving snake. In some games, the end of the trail is in a fixed position, so the snake continually gets longer as it moves. In another common scheme, the snake has a specific length, so there is a moving tail a fixed number of units away from the head. The player loses when the snake runs into the screen border, other obstacle, or itself.

4. Requirement Analysis

Project analysis is the process of collecting and analyzing the requirements of the project to be developed in order to recognize the need and objective of student and children to play the game for a fun.

C language

Software is a collection of program and a program is a collection of instruction given to the computer. Development of software is a stepwise process, before developing a software number of processes are down. The first step is to understand the used requirements. Problems analysis arises during the requirement phase of software development. C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972.It is a very popular language, despite being old. C is strongly associated with UNIX, as it was developed to write the UNIX operating system. It is a basic general purpose programming language. It is the base of all high level programming language so it is called basic programming language.

Features of C programming language

1. Simple
2. Powerful
3. Portable
4. Machine Independent
5. Structure Oriented
6. Middle level programming language
7. High speed
8. High Efficiency
9. Flexible

Application of C language

1. Developing Game
2. Operating System
3. Compilers
4. Editors
5. Database System
6. Network Drivers
7. Graphics packages
8. Interpreters

#

It includes the library of c into the program before the execution of program. It is also called preprocessor.

Include.h

To include the header file into the program.

Conio.h

The stand for consol input output. It is used show the output on console windows.

Void

It is a keyword. It indicates that no one value is being returned by the function. If we use any other keyword like int, float, char etc in place of void then we will use return keyword.

Main

It is the function which is called the entry point of any program. The execution of any prgram starts from the main function. If in a program there is only one function then it should be main function.

**Clrscr()**

It stands for clear screen. It is a predefined function which is used to clear the output screen. It acts like a duster on output screen. It is defined in the conio.h header file.

**Printf()**

It is a predefined function which is use used to print information or data on to the output screen. It is defined in the stdio.h header file.

**Getch()**

It is a predefined function which is used to hold the output screen. 2.It acts like a duster on output screen. 3.It is defined in the conio.h header file.

a. Hardware and software requirement

It indicates hardware and software requirement of the project at the time of project development as well as project implementation.

Minimum hardware requirements for efficient operation of this project are as follows:

1. Processor :Pentium and above
2. Hard disk :10MB or above
3. RAM :128 MB

iv. Output Unit :Monitor

1. Input Unit :Keyboard

Minimum Software requirements for efficient operation of this project are as follows:

1. Operating System Windows 7 and above version

**5.Algorithm in C Language**

Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output. Algorithms are generally created independent of underlying languages, i.e. an algorithm can be implemented in more than one programming language.

**From the data structure point of view, following are some important categories of algorithms:-**

* **Search** − Algorithm to search an item in a data structure.
* **Sort** − Algorithm to sort items in a certain order.
* **Insert** − Algorithm to insert item in a data structure.
* **Update** − Algorithm to update an existing item in a data structure.
* **Delete** − Algorithm to delete an existing item from a data structure.

## Characteristics of an Algorithm

Not all procedures can be called an algorithm. An algorithm should have the following characteristics :-

* **Unambiguous** − Algorithm should be clear and unambiguous. Each of its steps (or phases), and their inputs/outputs should be clear and must lead to only one meaning.
* **Input** − An algorithm should have 0 or more well-defined inputs.
* **Output** − An algorithm should have 1 or more well-defined outputs, and should match the desired output.
* **Finiteness** − Algorithms must terminate after a finite number of steps.
* **Feasibility** − Should be feasible with the available resources.
* **Independent** − An algorithm should have step-by-step directions, which should be independent of any programming code

6. Flowchart

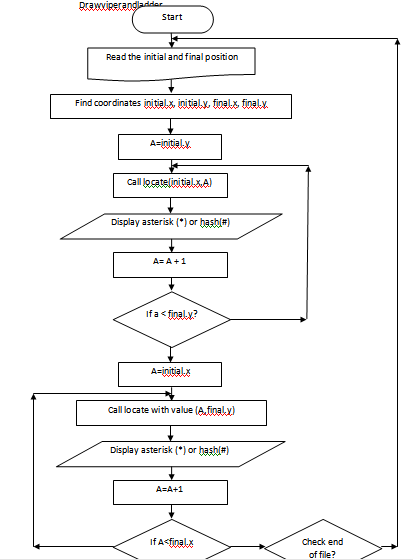
****

Fig :- flowchart

7.Coding

The entire coding of the project is done in Dev c c++ as the project is based on the C language. The entire coding has been already pasted in the above pages.

#include<stdio.h>

#include<stdlib.h>

#include<stdbool.h>

boolgameOver;

bool hit = false;

const width = 50;

const height = 20;

intx,y,fruitX,fruitY,score;

inttailX[200],tailY[200];

intntail=3;

char snake[200];

typedefenum {STOP = 0,LEFT,RIGHT,UP,DOWN} Direction;

Direction Dir;

void menu(){

int a;

printf("\n\n\n\n");

printf(" WEL COME TO DIPCHAN LAGARWAR ");

printf("\n\n")

printf(" THE SNAKE GAME\n ");

printf("1 )EW GAME \n ");

printf("2 GAME WITH BORDERS \n");

printf("3. QUIT THE GAME \n");

printf(" how may i help players \n");

printf(" Please enter your choice:");

scanf("%d", &a);

}

// This function is used to initialize(reset the game when it begins

void setup(){

// This function will be used to print the snake in each of its different cases inside its fame

gameOver = false;

Dir = STOP;

x= width/2;

y= height/2;

fruitX = rand()%width;

fruitY = rand()%height;

score = 0;

}

void draw(){

// This is used to take inputs when we press on any key on the keyboard and send it to the logic function to be interpreted

// This is used to

system("cls");

printf("Score:%d",score);

printf("\n");

inti;

for(i= 0 ;i<width+1;i++){ // first row

printf("=");

}

printf("\n");

int p;

for(p= 0 ;p<height;p++){

int q;

for(q= 0 ;q<width;q++){

if(q==0 || q==width-1){ // first last elt

printf("!");

}

if(p == y && q == x){//head coordinates

printf("\*");

}else if(p == fruitY&& q == fruitX){

printf("#");

}else{

int k=0;

bool print = false;

for(k=0;k<ntail;k++){

if(tailX[k]==q &&tailY[k]==p){

printf("\*");

print = true;

}

}

if(!print){printf(" ");}

}

}

printf("\n");

}

int j;

for(j= 0 ;j<width+1;j++){

printf("=");

}

}

void input(){

// The logic function

if(\_kbhit()){

switch(\_getch()){

case '4':

Dir = LEFT;

hit= true;

break;

case '8':

Dir = UP;

hit= true;

break;

case '6':

Dir = RIGHT;

hit= true;

break;

case '2':

Dir = DOWN;

hit= true;

break;

case 'x':

gameOver = true;

break;

}

}else if(!hit){

x++;

}

}

void logic(){

// main function

intlastX = tailX[0]; // The one before 0

intlastY = tailY[0];

int last2X, last2Y;

tailX[0]=x;

tailY[0]=y;

inti=0;

for(i=1; i<ntail;i++){

last2X = tailX[i];// 2 takes place of 1

last2Y = tailY[i];

tailX[i]=lastX; // 1 takes place of 0

tailY[i]=lastY;

lastX = last2X; // 1 takes place of 1

lastY = last2Y;

}

switch(Dir){

case UP:

y--;

break;

case DOWN:

y++;

break;

case LEFT:

x--;

break;

case RIGHT:

x++;

break;

}

if(x<0 || width<x || y<0 || height<y){

gameOver = true;

system("cls");

printf("#\*\*\*\*\*\*#######################\*\*\*\*# GAME OVER #\*\*\*#######################\*\*\*\*\*\*#");

}

if(x==fruitX&& y==fruitY){

ntail++;

score+=10;

fruitX = rand()%width;

fruitY = rand()%height;

}

}

int main(){

menu();

setup();

draw();

while(!gameOver){// each time we repeat a new picture will new drawn and we will see the snake in its new state

draw();

input();

logic();

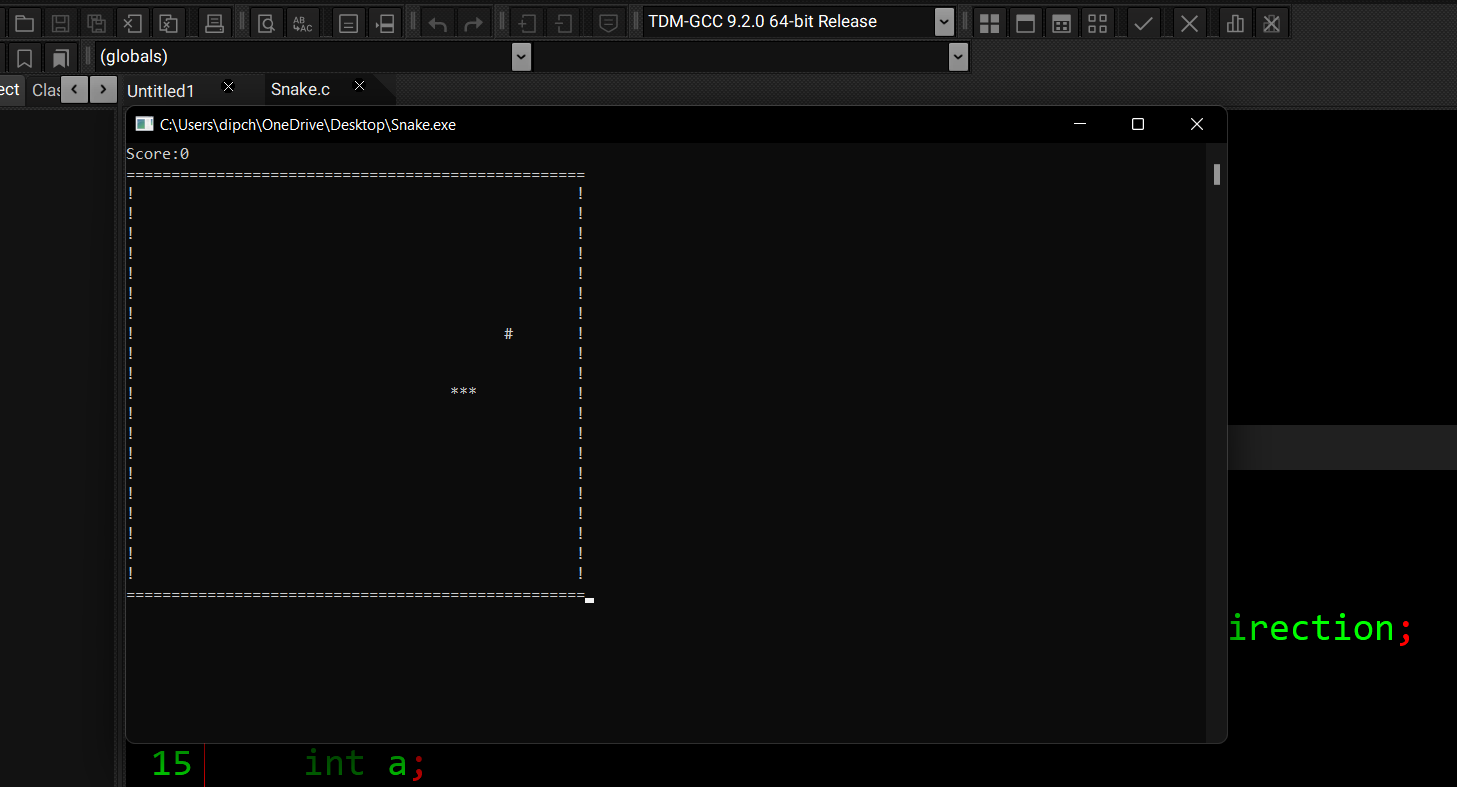
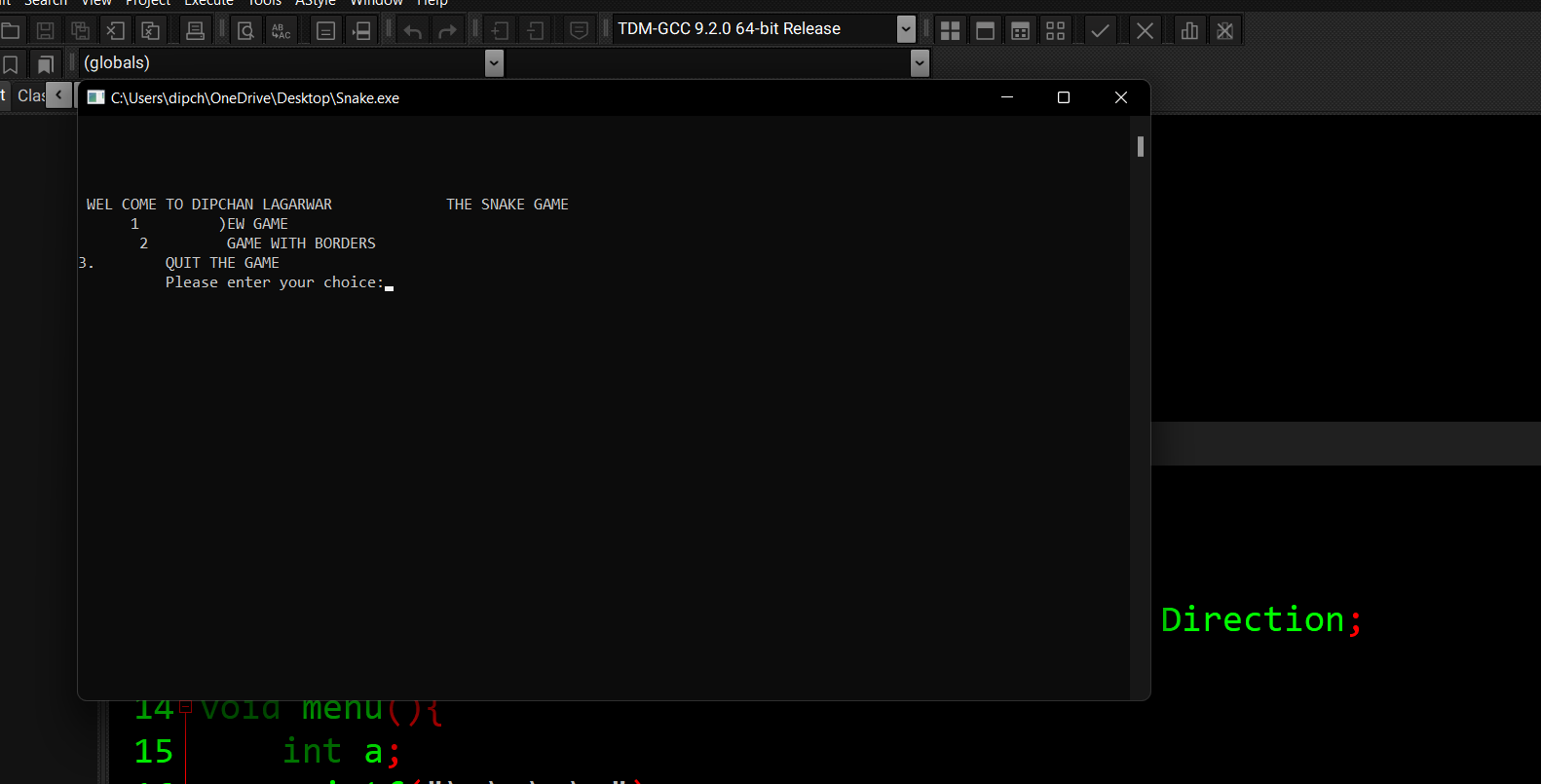
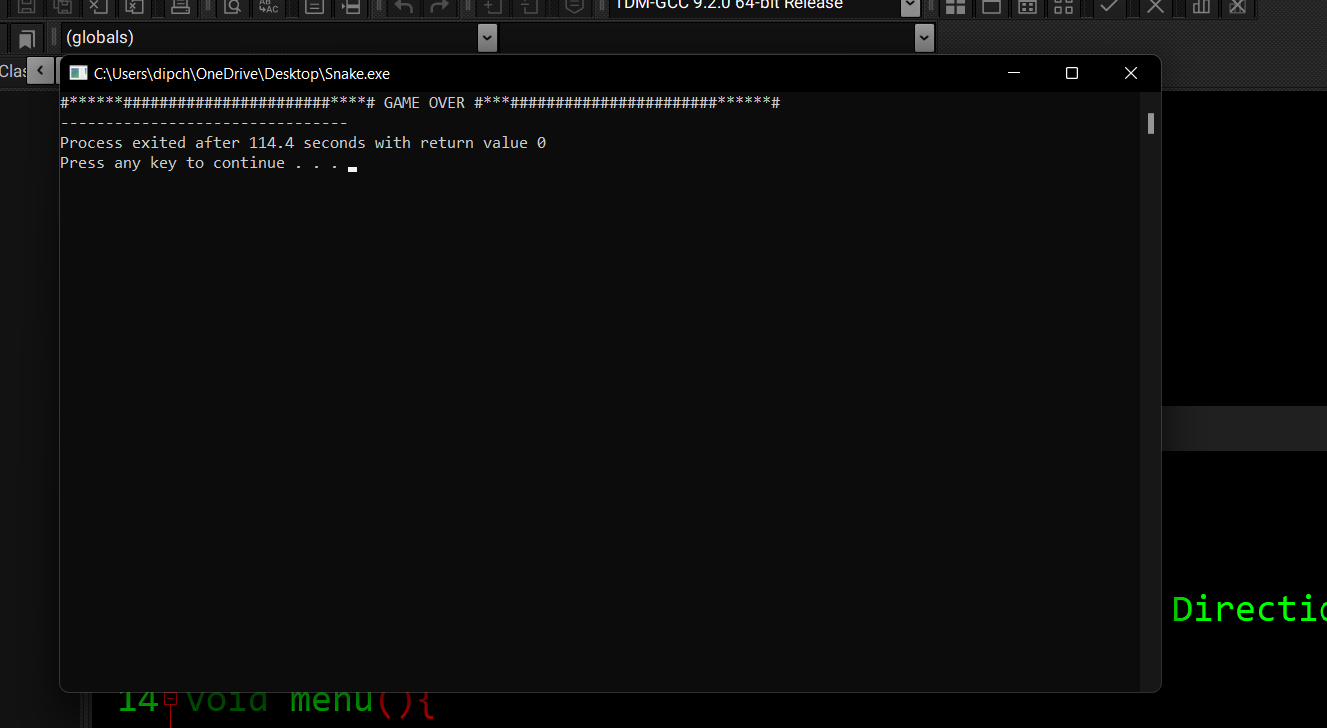
}

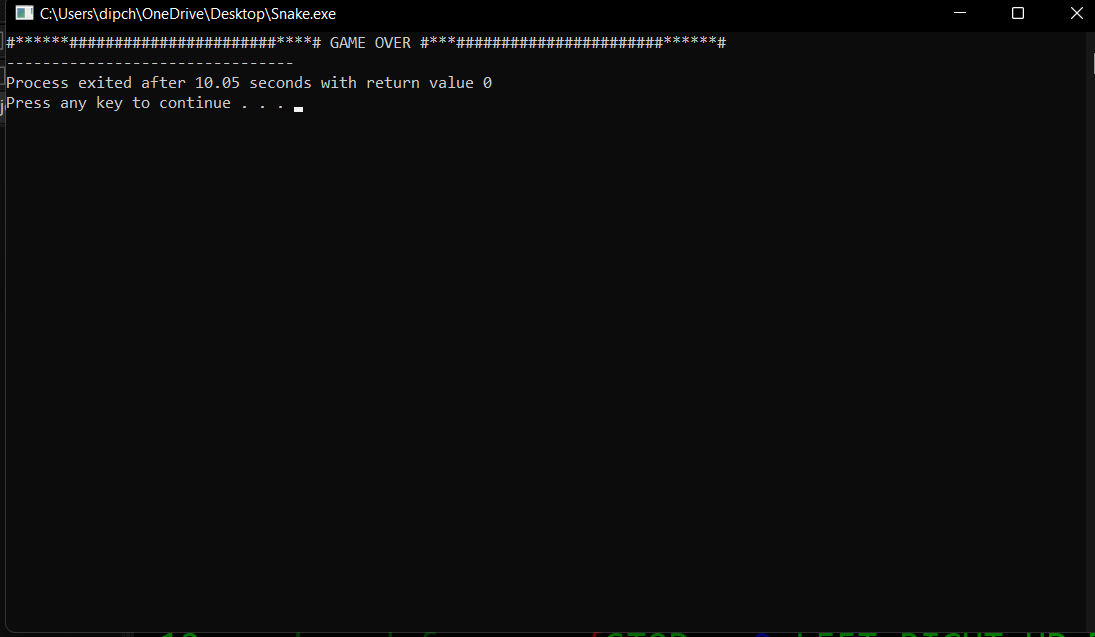
return 0;

}

8.Testing and output

Testing in a project development is a very important task to find out the possible mistakes made by the developers. The project contains no error at all.This project has checked and the screenshots are given below:-





9. Implementation

It is the process of using the project in client’s computer. After the execution file has been created , this project can be copied from saved source to any secondary storage device and pasted to the required system. The project can be operated by opening it ,completely replacing the existing manual system.

10.Maintenance

Due to the change in time, requirements of the organization also change and this project can no longer fulfill its requirements. The change is necessary to keep the project running and maintenance may be required when the college changes its requirements.

11. Conclusion

The coding of Snake game was extremely difficult with many errors arising. Many system had to be written numerous ways before a final working a final working solution was found. For example, two different movement methods were used prior a final version however even the final version is flawed as vertical movement cause the snake to change scale. There were also issue with the food snake collision detection. While the final version resulted in a snake that could eat food the movement glitch caused the food to cause further size issues.

12. Refrences

* Yashvant Kanetker ‘Let us C’
* Bryons S. Gottfried ‘Programming with C’
* E. Balagurusamy ‘Programming in ANSI C’